A 3D parallel Particle-In-Cell solver for wave interaction with vertical cylinders

Qiang Chen[a,*], Jun Zang[a], David M. Kelly[b], Aggelos S. Dimakopoulos[c]

[a] *Research Unit for Water, Environment and Infrastructure Resilience (WEIR), Department of Architecture and Civil Engineering, University of Bath, BA2 7AY, U.K.*
[b] *Coastal Research Lab., International Hurricane Research Center, Florida International University, Miami, Florida, 33199, U.S.A*
[c] *HR Wallingford, Wallingford, Oxon, OX10 8BA, U.K.*

Abstract

In this paper, the Particle-In-Cell (PIC) based PICIN solver is extended to three spatial dimensions and parallelised using the Message Passing Interface (MPI) approach. The PICIN solver employs both Eulerian grid and Lagrangian particles to solve the incompressible Navier-Stokes equations for free-surface flows. The particles are employed to carry all the fluid properties, solve the non-linear advection term and track the free-surface, while the grid is used solely for computational efficiency in solving the non-advection terms. Validation of the new 3D model concentrates on test cases involving multiple wave types (including regular waves, focused waves and solitary waves) interacting with vertical cylinders in several spatial configurations. The results are compared with laboratory data and numerical results from state-of-the-art Volume of Fluid (VOF) based Eulerian solvers such as those from the OpenFOAM® suite. It is shown that the 3D parallel PICIN model is able to well simulate highly non-linear water waves, and the interaction of such waves with vertical cylinders, with a CPU efficiency similar to Eulerian solvers. Moreover, the innovative use of particles in PICIN, akin to meshless Lagrangian solvers, gives the model a particular flexibility in handling complex, full 3D, water-wave scenarios involving large free-surface deformations.

Keywords: Particle-In-Cell, hybrid Eulerian-Lagrangian method, wave-structure interaction, Computational Fluid Dynamics, SPH, MPI parallelisation

*Corresponding author.
Email address: chenqiang913@hotmail.com (Qiang Chen)

1. Introduction

The Particle-In-Cell (PIC) method was first invented at the Los Alamos National Laboratory in 1955 by Harlow (1955) and was further developed there until it became the practical methodology described in Harlow (1964). The PIC method was designed in an attempt to combine the advantages of Eulerian and Lagrangian methods, through a combined use of grid and particles. In particular, the particles are effectively used to solve the transport of fluid properties in a Lagrangian manner, while other non-advection terms are resolved on the underlying grid. In Harlow (1964), the fluid properties such as the mass and momentum are split between particles and grid; the velocity field needs to be transferred back and forth between particles and grid and thus it makes the solver highly dissipative. Following that, further developments of the PIC method can be found in, for example, Brackbill and Ruppel (1986) and Brackbill et al. (1988). In their PIC solver, the particles are assigned with all the fluid properties, and the grid is solely used for computational convenience in solving the governing equations for the velocity change (i.e. acceleration) on the grid, which is then used to increment particle velocity. This reduces the numerical dissipation significantly. The current 3D parallel PICIN solver follows the latter approach.

The PIC method was originally intended for compressible fluid flows and was extended to solve for incompressible flows by Harlow and Welch (1965)(via the Marker-and-Cell method which employs Eulerian advection) and Zhu and Bridson (2005). More recently, the PIC method has been developed by using innovative techniques, enabling the method to successfully model fluid-structure interaction processes in the coastal and offshore environment. Initially, Kelly (2012) developed a PIC-based solver to simulate the propagation and breaking of a solitary wave on an idealized beach. Kelly et al. (2015) then developed the PICIN solver for free-surface flows with fluid-solid interaction using a tailored Distributed Lagrange Multiplier (DLM) method. Chen et al. (2015) applied the PICIN solver to simulate the complex industrial problem of rock dumping through fall-pipes. Chen et al. (2016a) further validated the PICIN solver for 2D coastal flows. Later, Chen et al. (2016b) suggested that the DLM method used in PICIN may require further developments for simulating surface-piercing floating structures. They instead modified the PICIN model by incorporating a Cartesian cut cell based two-way strong coupling algorithm for fluid-structure interactions. In their study, the modified PICIN model was demonstrated to be capable of handling violent free-surface flows as well as the interaction of water waves with (surface-piercing) floating structures of arbitrary shapes and degree of freedoms.

In the studies mentioned above, the PICIN model was tested only in two spatial dimen-

sions, showing nevertheless a great potential to be used as a numerical tool over a range of practical engineering applications. Extending PICIN to three spatial dimensions comes with additional requirements for computational resources due to the highly demanding memory storage, as a double grid system (i.e. particle and grid) is employed. To mitigate this problem, this paper chooses to parallelise the 3D PICIN model using the memory distribution based Message Passing Interface (MPI) approach, such that the model can make use of the High Performance Computing Service (HPCS). The major components of the parallelisation of the 3D PICIN model are detailed in this paper.

The validation tests are based on wave interaction with vertical cylindrical structures, which are widely employed in coastal and offshore engineering. Examples include the design of oil platforms, offshore wind turbine foundations and piled wharfs (Zhu and Moule, 1996). Numerous studies, both numerical and experimental, have been carried out on this topic. For example, Chen et al. (2014) employed the OpenFOAM® model to study the nonlinear effects of both regular wave and focused wave interaction with a single cylinder. This case was experimentally studied by Zang et al. (2010). Kamath et al. (2015) simulated regular wave interaction with multiple cylinders placed in groups in different configurations using the open-source CFD model REEF3D. Mo and Liu (2009) investigated non-breaking solitary wave interaction with a single cylinder or a group of three cylinders through a Volume of Fluid (VOF) based finite volume numerical solver and also physical experiments. Lara et al. (2013) and Leschka and Oumeraci (2014) also investigated solitary wave interaction with three vertical cylinders based on OpenFOAM®; in both works, different configurations of the cylinders are studied. In addition to the above-mentioned Eulerian methods, Lagrangian methods have also been widely employed for investigating these topics, such as the Smoothed Particle Hydrodynamics (SPH) method (Dalrymple and Rogers, 2006; Lind et al., 2016) and the meshless local Petrov-Galerkin method based on Rankine source solution (MLPG_R) (Zhou et al., 2009).

This paper aims to contribute to further understanding the processes of the interaction between different waves and single or multiple vertical cylinders, in terms of wave elevations around the cylinders, wave run-up and wave loading. More importantly, attention has been focused on the comparison of results between the PICIN model and VOF-based Eulerian models such as OpenFOAM®. We show that the innovative use of particles to track the free surfaces in PIC can handle large free-surface deformations such as wave breaking more advantageously. We also demonstrate that with the MPI parallelisation the PIC-based PICIN model is capable of modelling large scale 3D water wave problems, with an efficiency

(in terms of CPU cost) that is of the same magnitude as the OpenFOAM® model.

The paper is organised as follows: Section 2 gives an overview of the 3D parallel PICIN model including the governing equations, major numerical implementations and the parallelisation using the MPI approach. Next, Section 3 compares the results of the present model with experimental measurements and numerical predictions from other state-of-the-art numerical solvers for two idealised test cases. Finally, in Section 4 conclusions are drawn.

2. The 3D PICIN Model

2.1. Governing equations

The PICIN model solves the incompressible Newtonian Navier-Stokes equations for single-phase flows:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \boldsymbol{f} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u}, \tag{2}$$

where, in 3D, $\boldsymbol{u} = [u, v, w]^T$ is the velocity field; t is time; p is pressure; $\boldsymbol{f}$ = [0.0, 0.0, -9.81 m/s^2]^T represents the body force due to gravity, and ρ and ν are the density and kinematic viscosity of the fluid, respectively. According to the PIC methodology, both grid and particles are employed to solve the governing equations. In particular, a staggered grid is employed following Harlow and Welch (1965), where pressures are stored at cell centres, whose positions along the x-, y- and z-directions are numbered by indexes i, j and k, respectively, and velocities are computed at relevant cell faces, whose positions are labelled with half-integer values of the indexes. Fig. 1 shows a schematic of the computational setup, where the staggered grid and fluid particles are also depicted. In the current 3D PICIN model, 8 particles are initially seeded in each cubic cell. Cells occupied by the particles are marked as fluid cells, while air cells correspondingly have no particles inside. The particles carry the fluid properties such as the mass and momentum, and are used to track the configuration of the fluid body (including the free-surface position) and solve the nonlinear advection term (the second term on the left hand side of the momentum equation), while the underlying grid is employed solely for computational convenience for solving the non-advection terms. The solution procedure is divided into two major steps: an Eulerian step and a Lagrangian step. During the Eulerian step the governing equations, ignoring the nonlinear advection term, are resolved on the grid. After that, in the Lagrangian step, the solution on the grid is used to update the particle velocity and the remaining advection term is then handled using the particles in a Lagrangian manner. More details

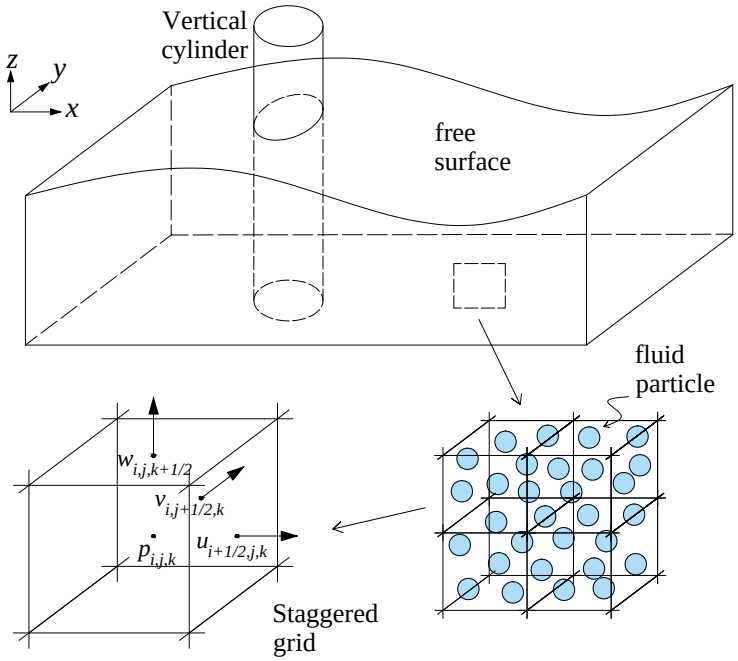

Fig. 1: Sketch of the computational domain, the staggered grid and fluid particles.

of the solution procedure are given in the following sections. We note that no turbulence models are incorporated in the current numerical model, and thus the test cases presented in Section 3 are carefully selected.

2.2. Eulerian step

2.2.1. The pressure projection algorithm

In the Eulerian step, the governing equations, ignoring the nonlinear advection term in the momentum equation, are solved on the grid. Prior to the solutions, the velocity field $\boldsymbol{u}^n$ at the nth time-step on the grid is mapped from the velocity field carried by the particles; this is discussed in Section 2.4.

The solution in this step uses the pressure projection method proposed in Chorin (1968). A tentative velocity $\tilde{\boldsymbol{u}}$ is first computed by applying the body force and physical viscosity term as an Euler step in time:

$$\frac{\tilde{\boldsymbol{u}} - \boldsymbol{u}^n}{\Delta t} = \nu \nabla^2 \boldsymbol{u}^n + \boldsymbol{f}, \tag{3}$$

where Δt is the time step. The body force and the viscosity term are both treated explicitly on the grid, and the viscosity term is resolved using the central differencing method. The

next step is then to find a pressure field p^{n+1} to maintain the incompressibility condition (Eq. 1). Recalling the remaining pressure gradient part of the momentum equation, we have:

$$\frac{(\boldsymbol{u}^{n+1} - \tilde{\boldsymbol{u}})}{\Delta t} = -\rho^{-1}\nabla p^{n+1}. \tag{4}$$

Taking the divergence of both sides of Eq. 4 and recalling that the velocity at the next time-step must satisfy the divergence-free condition lead to a pressure Poisson equation (PPE):

$$\Delta t \rho^{-1}\nabla^2 p^{n+1} = \nabla\cdot\tilde{\boldsymbol{u}}\,. \tag{5}$$

The PPE is discretised and solved in a finite volume sense by integrating both sides over the computational cells and evoking the divergence theorem. Also, during the solution of the PPE, the boundary conditions on both the free surface and the structure surface are resolved. A discretisation of the PPE incorporating the boundary conditions is discussed in Section 2.2.2. Once the pressure field p^{n+1} is found, the next step is to project the tentative velocity field $\tilde{\boldsymbol{u}}$ onto a divergence-free velocity field using the pressure as a Lagrange multiplier:

$$\boldsymbol{u}^{n+1} = \tilde{\boldsymbol{u}} - \Delta t \rho^{-1}\nabla p^{n+1}. \tag{6}$$

2.2.2. Boundary conditions and solution of the PPE

The boundary conditions for the structure surface and the free surface are resolved during the solution of the PPE. On the structure surface, the boundary conditions are imposed following Ng et al. (2009):

$$\boldsymbol{n}\cdot\boldsymbol{u} = \boldsymbol{n}\cdot\boldsymbol{U}_b \quad \text{and} \quad \boldsymbol{n}\cdot(\Delta t\rho^{-1}\nabla p) = \boldsymbol{n}\cdot(\tilde{\boldsymbol{U}}_b - \boldsymbol{U}_b^{n+1}) \quad \text{on } \partial\Omega_S(\boldsymbol{x},t), \tag{7}$$

where $\partial\Omega_S$ represents the structure surface; $\boldsymbol{n}$ is the unit outward normal vector of the structure surface; $\tilde{\boldsymbol{U}}_b$ represents a tentative velocity on the structure surface; $\boldsymbol{U}_b^{n+1}$ is the velocity imposed on the structure surface at time level $n+1$. The boundary conditions on the structure surface are applied using the Cartesian cut cell based fluid-structure coupling algorithm proposed in Ng et al. (2009). Integrating the PPE (Eq. 5) over a fluid cell, G_{ijk}, that is partially occupied by solid structures and evoking the divergence theorem and the

above boundary conditions, we obtain:

$$
\begin{aligned}
E_{i-\frac{1}{2},j,k}\cdot\frac{\Delta t(p^{n+1}_{i-1,j,k}-p^{n+1}_{i,j,k})}{\rho\Delta x}+E_{i+\frac{1}{2},j,k}\cdot\frac{\Delta t(p^{n+1}_{i+1,j,k}-p^{n+1}_{i,j,k})}{\rho\Delta x}+\\
E_{i,j-\frac{1}{2},k}\cdot\frac{\Delta t(p^{n+1}_{i,j-1,k}-p^{n+1}_{i,j,k})}{\rho\Delta y}+E_{i,j+\frac{1}{2},k}\cdot\frac{\Delta t(p^{n+1}_{i,j+1,k}-p^{n+1}_{i,j,k})}{\rho\Delta y}+\\
E_{i,j,k-\frac{1}{2}}\cdot\frac{\Delta t(p^{n+1}_{i,j,k-1}-p^{n+1}_{i,j,k})}{\rho\Delta z}+E_{i,j,k+\frac{1}{2}}\cdot\frac{\Delta t(p^{n+1}_{i,j,k+1}-p^{n+1}_{i,j,k})}{\rho\Delta z}\\
=E_{i+\frac{1}{2},j,k}\cdot\tilde{u}_{i+\frac{1}{2},j,k}-E_{i-\frac{1}{2},j,k}\cdot\tilde{u}_{i-\frac{1}{2},j,k}+E_{i,j+\frac{1}{2},k}\cdot\tilde{v}_{i,j+\frac{1}{2},k}-E_{i,j-\frac{1}{2},k}\cdot\tilde{v}_{i,j-\frac{1}{2},k}+\\
E_{i,j,k+\frac{1}{2}}\cdot\tilde{w}_{i,j,k+\frac{1}{2}}-E_{i,j,k-\frac{1}{2}}\cdot\tilde{w}_{i,j,k-\frac{1}{2}}-\int_{G_{ijk}\bigcap\partial\Omega_S}\boldsymbol{n}\cdot\boldsymbol{U}_b^{n+1}\,dA\,, \quad (8)
\end{aligned}
$$

where the subscripts are the space indexes as mentioned in Section 2.1; E represents the fraction of a cell face that is open-to-water (i.e. not occupied by structures); $\Delta x=\Delta y=\Delta z$ are the grid sizes and dA refers to the area differential. For the detailed process of the derivation of Eq. 8, the reader is referred to Ng et al. (2009). It is noted that since only fixed and motion prescribed structures (e.g. wavemaker) are investigated in this paper, $\boldsymbol{U}_b^{n+1}$, the velocity imposed on the structure surface, in the last term on the right hand side of Eq. 8 is known. The resulted linear system of equations are solved using the bi-conjugate gradient (BCG) method (Press et al., 1992), which is parallelised using the MPI approach for the current model; details are given in Section 2.7.

On the free surface, the boundary condition enforced is:

$$p=0 \quad \text{on } \zeta(\boldsymbol{x},t), \tag{9}$$

where $\zeta(\boldsymbol{x},t)$ represents the free-surface position reconstructed on the grid based on the particle position. That is, following Chen et al. (2016b), a signed distance function (SDF), ϕ, is constructed at the cell centres at each time-step such that $\phi>0$ represents the atmospheric pressure region, $\phi<0$ denotes the fluid region, and $\phi=0$ corresponds to the free-surface position. The free-surface boundary condition is imposed using the second-order accurate technique proposed in Gibou et al. (2002) in a dimension by dimension manner based on the SDF ϕ. The main idea is to linearly extrapolate an artificial pressure from the fluid cell with $\phi\leq 0$ to its adjacent cell with $\phi>0$ such that the pressure at the free-surface position is zero. For example, assuming that cell $(i+1,j,k)$ has $\phi_{i+1,j,k}>0$ and its adjacent cell (i,j,k) in the x-direction has $\phi_{i,j,k}\leq 0$, then the artificial pressure in cell $(i+1,j,k)$ is given

by:

$$p_{i+1,j,k} = (1 - \frac{1}{\Theta})p_{i,j,k}\,, \tag{10}$$

where Θ is calculated by:

$$\Theta = \frac{|\phi_{i,j,k}|}{\phi_{i+1,j,k} + |\phi_{i,j,k}|} + \lambda\,, \tag{11}$$

where λ in the current model is set to $\frac{0.001\Delta x}{\phi_{i+1,j}+|\phi_{i,j}|}$ for the purpose of ensuring numerical stability. The artificial pressure, i.e. Eq. 10, is substituted into Eq. 8 where applicable to reflect the presence of the free-surface boundary. We note that this only changes the coefficient matrix of the linear system of equations of Eq. 8.

2.3. Lagrangian step

In this step, the remaining nonlinear advection term is solved using the particles in a Lagrangian manner. As mentioned above, the underlying grid is solely employed for computational convenience for solving the non-advection terms in the governing equations. In particular, this is for computing the accelerations of fluid particles. Once the divergence-free velocity field $\boldsymbol{u}^{n+1}$ on the grid is found in the Eulerian step, the velocity change (i.e. the acceleration), $\boldsymbol{a}^{n+1} = \boldsymbol{u}^{n+1} - \boldsymbol{u}^{n}$, on the grid is interpolated onto the particles to increment the velocity carried by the particles. The interpolation uses the fourth-order accurate weighted essentially non-oscillatory (WENO) scheme proposed by Edwards and Bridson (2012). It is noteworthy that only incrementing the particle velocity using the velocity change allows the associated numerical errors to accumulate (Jiang et al., 2015), which can lead to numerical instability issues. In order to alleviate this problem, following Zhu and Bridson (2005), the divergence-free velocity itself on the grid is also interpolated to constitute a proportion of the final particle velocity in the current model. This in fact introduces some numerical dissipation as the simulation progresses, because the velocity field is interpolated back and forth between the particles and grid at each time step. The final particle velocity is thus calculated by:

$$\boldsymbol{v}_p^{n+1} = c(\boldsymbol{v}_p^{n} + \sum_i \boldsymbol{a}^{n+1} S_i) + (1 - c)\sum_i \boldsymbol{u}^{n+1} S_i\,, \tag{12}$$

where $\boldsymbol{v}_p$ is the particle velocity; S_i represents the WENO interpolation function; c is an empirical blending coefficient. The chosen value of c should fit the purpose of stabilising the code while introducing as little unwanted numerical dissipation as possible. In this paper, c is chosen to be 0.96 following Chen et al. (2016b).

Following Brackbill and Ruppel (1986), the particles, carrying the newly updated velocity field, are then advected through the divergence-free velocity field on the grid, rather

than the velocity field they carry. In other words, the particles are advected using the velocities directly interpolated from the divergence-free velocity field on the grid, i.e. the term $\sum_i \boldsymbol{u}^{n+1} S_i$ in Eq. 12. Also, in order to reduce numerical errors, the third-order accurate Runge-Kutta scheme of Ralston (1962) is employed for the particle advection. In this way, the nonlinear advection term is resolved straightforwardly, giving the PICIN solver a flexibility akin to meshless methods such as SPH. It is noted that this idea, that the velocity from the momentum equation and the velocity used to change the positions need not be identical, is also well exploited for the SPH method (Monaghan, 1989). After the particles are advected, one computational cycle of solving the Navier-Stokes equations is completed.

It is noted that in order to correctly update the velocities and positions for those particles around the free surface, the velocity field on the grid is extrapolated from the fluid region to the atmospheric pressure region a number of times during one computational cycle. The velocity extrapolation uses the Fast Marching Method proposed by Adalsteinsson and Sethian (1999) to solve equation $\nabla\phi \cdot \nabla q = 0$ in a dimension by dimension manner, where q refers to the velocity component, i.e. u, v, and w, on the grid. The velocity field is also extrapolated from the fluid region to the regions occupied by structures in a similar fashion for those particles around the structure surfaces (see more discussions in Section 2.6). Also, it is noteworthy that in the current 3D PICIN model a particle redistribution scheme following Kelly et al. (2015) is applied every 30 time-steps on those particles around the structure surfaces in order to maintain a relatively regular particle distribution and thus reduce unphysical voids around the structures, which could occur due to the inherent problem in PIC that the particle distribution tends to become irregular as the simulation progresses (Kelly et al., 2015). Moreover, the time step is controlled by the Courant number. For full details of the implementations of these techniques, the reader is referred to Chen (2017).

2.4. Data mapping from particles to grid

Prior to solving the governing equations on the grid, the data represented by the particles must be mapped onto the grid. As the particles carry all the fluid properties, the data mapping includes the fluid configuration (i.e. the free-surface position) and the fluid mass and momentum. For the former, in the current model, it is achieved by computing a signed distance function based on the particle position as mentioned in Section 2.2.2. For the latter,

the technique used here is akin to the SPH method:

$$m_g \boldsymbol{u}_g = \sum_{p \in \Omega_g} \boldsymbol{v}_p m_p \sigma_{pg}(\boldsymbol{x}_p - \boldsymbol{X}_g, k_h) \ , \tag{13}$$

$$m_g = \sum\nolimits_{p \in \Omega_g} m_p \sigma_{pg}(\boldsymbol{x}_p - \boldsymbol{X}_g, k_h) \ , \tag{14}$$

where the subscripts p and g represent indexes of a particle and a grid node, respectively; $\boldsymbol{u}_g$ and m_g thus denote the fluid velocity and mass at the grid node g; $\boldsymbol{v}_p$ and m_p are the fluid velocity and mass carried by the particle p; $\boldsymbol{x}_p$ and $\boldsymbol{X}_g$ are the position vectors for the particle p and the grid node g, respectively; σ_{pg} can be interpreted as a fraction of the mass, as well as the momentum, of particle p that are assigned to the grid node g during this data mapping, and it is computed by:

$$\sigma_{pg}(\boldsymbol{x}_p - \boldsymbol{X}_g, k_h) = \frac{W_p(\boldsymbol{x}_p - \boldsymbol{X}_g, k_h)}{\sum_{\tilde{g} \in \Omega_p} W_p(\boldsymbol{x}_p - \boldsymbol{X}_{\tilde{g}}, k_h)} \ , \tag{15}$$

where W_p is the cubic spline kernel function (Monaghan and Lattanzio, 1985), with particle p being the reference point; k_h is the so-called kernel length; Ω_p and Ω_g denote the spherical support domains in three spatial dimensions, with particle p and grid node g located at the centres respectively and the radius both being $2k_h$; $\tilde{g}$ represents the grid node whose velocity component is in the same direction as that of the grid node g. It is noted that because a staggered grid is employed, Eq. 13–Eq. 15 are calculated in a dimension by dimension manner. Also, Eq. 13 and Eq. 14 conserve the mass and momentum during this mapping procedure, due to the fact that in any direction $\sum_{g \in \Omega_p} \sigma_{pg} = 1$. Taking the mass mapping in one direction as an example, this can be proved following Jiang et al. (2015):

$$\begin{aligned} \sum\nolimits_g m_g &= \sum\nolimits_g \sum\nolimits_{p \in \Omega_g} m_p \sigma_{pg} \\ &= \sum\nolimits_p m_p (\sum\nolimits_{g \in \Omega_p} \sigma_{pg}) \\ &= \sum\nolimits_p m_p \ . \end{aligned} \tag{16}$$

2.5. Numerical wave tank

A numerical wave tank (NWT) is established in the current 3D PICIN model, which is basically an extension of the 2D NWT described in Chen et al. (2016a). For the wave generation, a piston-type wavemaker is employed at one end of the NWT and is resolved using the Cartesian cut cell based fluid-structure coupling algorithm discussed in Section 2.2.2.

The piston-type wavemaker moves along the x-direction in the current 3D NWT and the displacement and velocity of the wavemaker are determined according to the linear wavemaker theory (Dean and Dalrymple, 1991) and the type of waves under consideration. In particular, for the generation of focused wave and solitary wave, the methods discussed in Zhao and Hu (2012) and Wu et al. (2016) are adopted to determine the wavemaker velocity, respectively.

For the wave absorption, the relaxation zone approach of Jacobsen et al. (2012) is adopted. The velocity carried by the particles, entering the absorption zone at the downstream end of the NWT in the x-direction, is forced to be the desired analytical solution, which in our case is water at rest:

$$\boldsymbol{v}_r = \xi(x_p)\boldsymbol{v}_{ana} + (1 - \xi(x_p))\boldsymbol{v}_{num}, \tag{17}$$

where $\boldsymbol{v}_{ana} = [0, 0, 0]^T$ is the desired analytical velocity and $\boldsymbol{v}_{num}$ is the computed numerical velocity $[v_x, v_y, v_z]^T$, where v_x, v_y and v_z are the velocity components in the x-, y- and z-directions, respectively; $\xi(x_p)$ is a relaxation coefficient, and x_p is the x-component of the particle coordinate:

$$\xi(x_p) = 1 - \frac{\exp(x_r^{3.5}) - 1}{\exp(1) - 1}, \tag{18}$$

where $x_r = (x_p - x_{sta})/D_L$, and x_{sta} and D_L are the starting location and the length of the absorption zone in the x-direction, respectively.

Due to the setup of the wavemaker and wave absorption zone at both ends of the NWT in the x-direction, problems that involve currents cannot be simulated in the present 3D NWT. Nevertheless, for simulating currents using the PIC method, the reader is referred to Kelly et al. (2015) where a 2D test case involving flow past a circular cylinder close to a free surface is presented.

2.6. Numerical algorithm

The major components of the solution procedure of the 3D PICIN model presented in this paper is summarised below. The primary variables (velocity on the grid, velocity change on the grid, particle velocity and particle position) adjusted at each sub-step are also indicated.

(1) Move the wavemaker according to the wave generation method; $\mathbf{0}$, $\mathbf{0}$, $\boldsymbol{v}_p^n$, $\boldsymbol{x}_p^n$.

(2) Map the mass and momentum and the free-surface position from the particles to the grid; $\boldsymbol{u}^n$, $\mathbf{0}$, $\boldsymbol{v}_p^n$, $\boldsymbol{x}_p^n$.

(3) Extrapolate the velocity field from the fluid region to the atmospheric pressure region

and regions occupied by fixed structures; $\boldsymbol{u}^n$, $\mathbf{0}$, $\boldsymbol{v}_p^n$, $\boldsymbol{x}_p^n$.

(4) Solve the Navier-Stokes equations (ignoring the nonlinear advection term) on the grid; $\boldsymbol{u}^{n+1}$, $\mathbf{0}$, $\boldsymbol{v}_p^n$, $\boldsymbol{x}_p^n$.

(5) Extrapolate the velocity field from the fluid region to the atmospheric pressure region and regions occupied by fixed structures; $\boldsymbol{u}^{n+1}$, $\mathbf{0}$, $\boldsymbol{v}_p^n$, $\boldsymbol{x}_p^n$.

(6) Calculate the velocity change on the grid; $\boldsymbol{u}^{n+1}$, $\boldsymbol{a}^{n+1}$, $\boldsymbol{v}_p^n$, $\boldsymbol{x}_p^n$.

(7) Interpolate $\boldsymbol{u}^{n+1}$ and $\boldsymbol{a}^{n+1}$ from the grid to the particles to update the particle velocity and advect the particles; $\boldsymbol{u}^{n+1}$, $\boldsymbol{a}^{n+1}$, $\boldsymbol{v}_p^{n+1}$, $\boldsymbol{x}_p^{n+1}$.

(8) Conduct wave absorption (if the wave absorption zone is used); $\boldsymbol{u}^{n+1}$, $\boldsymbol{a}^{n+1}$, $\boldsymbol{v}_p^{n+1}$, $\boldsymbol{x}_p^{n+1}$.

(9) Conduct particle redistribution (every 30 time-steps); $\boldsymbol{u}^{n+1}$, $\boldsymbol{a}^{n+1}$, $\boldsymbol{v}_p^{n+1}$, $\boldsymbol{x}_p^{n+1}$.

(10) Update the time step and repeat steps (1)–(10).

It is noted that in the above solution procedure the velocity extrapolation is repeated in steps (3) and (5) for the purpose of calculating the velocity change (step (6)) in the atmospheric pressure region and regions occupied by fixed structures, which is required by the fourth-order accurate WENO interpolation scheme (step (7)) as four-point stencils are required (Edwards and Bridson, 2012). We note that the fluid velocity is also extrapolated into the region occupied by the wavemaker in a slightly different manner. Since the wavemaker is a piston-type paddle that always vertically cuts the grid cells, the velocities inside the wavemaker region in the y- and z-directions are simply copied from the adjacent fluid cells, while the velocity in the x-direction is determined according to the wave generation methods discussed in Section 2.5. However, the velocity changes inside the wavemaker region in all directions are copied from the adjacent fluid cells. These are implemented between steps (6) and (7) in the current numerical model.

2.7. Parallelisation of the 3D PICIN solver

The 3D PICIN model employs the MPI approach for parallelisation. The major components are introduced here. The computational domain is decomposed into a number of sub-domains as shown in Fig. 2. The information transfer (e.g. particle and grid properties) in the ghost cells is conducted using the MPI approach. In most cases, one layer of ghost cells are defined at the boundaries of each sub-domain. In certain instances, for example, when computing the fourth-order WENO scheme for grid-to-particle interpolations, one additional layer of ghost cells are used, as four-point stencils are required (Edwards and Bridson, 2012).

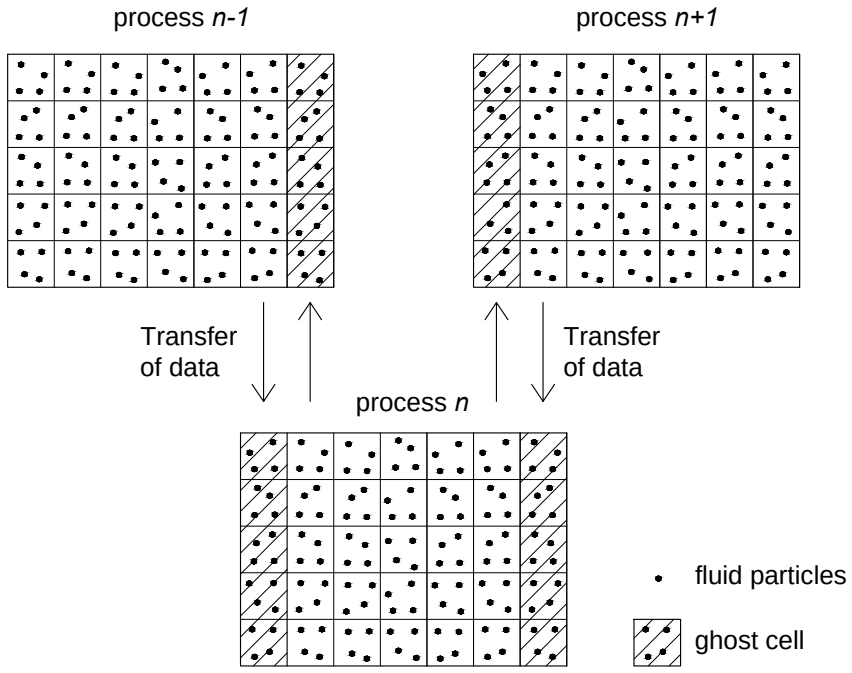

Fig. 2: Schematic (side view) showing the domain decomposition and data transfer via ghost cells in the 3D parallel PICIN model.

As mentioned above, the serial BCG solver of Press et al. (1992) is parallelised using the MPI approach in the PICIN model. Given a linear system of equations $\mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{b}$ with preconditioner $\tilde{\mathbf{A}}$ (the diagonal part of $\mathbf{A}$), Fig. 3 presents the algorithm for the parallelised BCG solver. It can be seen from Fig. 3 that the BCG algorithm requires several Matrix-Vector multiplications and Vector-Vector products. This is where data communication is required, as the matrix $\mathbf{A}$ is divided into the sub-domains. In Fig. 3, this is marked by bullet points.

The performance of parallelisation in the present model is tested through the simulation of solitary wave propagation in a NWT as described in Section 3.2. For the performance test here, a total number of 22.33 million particles and 7.78 million cells were used (note that in this test case while the water depth was 0.75 m, the height of the NWT was 2 m). The simulation was run on the Bath HPCS, which is based on the Intel Xeon E5-2650v2 IvybridgeV2 processor. Fig. 4 presents the trends of speedup (S_p) and efficiency (E_p), where

$$S_p = \frac{T_k}{T_p} \quad \text{and} \quad E_p = \frac{S_p}{p/k}\,; \tag{19}$$

T_p and T_k represent the CPU time, with p and k being the number of cores. Note that, ideally,

$\mathbf{x}_1$ = Initial guess

$\mathbf{r}_1 = \mathbf{b} - \mathbf{A} \cdot \mathbf{x}_1$

- Update $\mathbf{r}_1$ in ghost cells using MPI_SENDRECV

$\bar{\mathbf{r}}_1 = \mathbf{r}_1$

Set $\mathbf{p}_1 = \mathbf{r}_1$ and $\bar{\mathbf{p}}_1 = \bar{\mathbf{r}}_1$

Repeat k = 1,2...

Set $\tilde{\mathbf{A}} \cdot \mathbf{z}_k = \mathbf{r}_k$ and $\tilde{\mathbf{A}}^T \cdot \bar{\mathbf{z}}_k = \bar{\mathbf{r}}_k$

- Sum up $\bar{\mathbf{r}}_k \cdot \mathbf{z}_k$ in all processes using MPI_ALLREDUCE(MPI_SUM)
- Update the products of $\mathbf{A} \cdot \mathbf{p}_k$ in ghost cells using MPI_SENDRECV
- Sum up $\bar{\mathbf{p}}_k \cdot \mathbf{A} \cdot \mathbf{p}_k$ in all processes using MPI_ALLREDUCE(MPI_SUM)

$$\alpha_k = \frac{\bar{\mathbf{r}}_k \cdot \mathbf{z}_k}{\bar{\mathbf{p}}_k \cdot \mathbf{A} \cdot \mathbf{p}_k}$$

$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A} \cdot \mathbf{p}_k$

- Update the products of $\mathbf{A}^T \cdot \bar{\mathbf{p}}_k$ in ghost cells using MPI_SENDRECV

$\bar{\mathbf{r}}_{k+1} = \bar{\mathbf{r}}_k - \alpha_k \mathbf{A}^T \cdot \bar{\mathbf{p}}_k$

- Sum up $\bar{\mathbf{r}}_{k+1} \cdot \mathbf{z}_{k+1}$ in all processes using MPI_ALLREDUCE(MPI_SUM)
- Sum up $\bar{\mathbf{r}}_k \cdot \mathbf{z}_k$ in all processes using MPI_ALLREDUCE(MPI_SUM)

$$\beta_k = \frac{\bar{\mathbf{r}}_{k+1} \cdot \mathbf{z}_{k+1}}{\bar{\mathbf{r}}_k \cdot \mathbf{z}_k}$$

$\mathbf{p}_{k+1} = \mathbf{z}_k + \beta_k \mathbf{p}_k$

$\bar{\mathbf{p}}_{k+1} = \bar{\mathbf{z}}_k + \beta_k \bar{\mathbf{p}}_k$

$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$

- Call MPI_BARRIER to keep all processes in the same pace

Until stopping criteria is reached

Fig. 3: An algorithm of BCG solver (Press et al., 1992) parallelised using the MPI approach. The bullet points indicate the intervention of parallelisation.

in the above equations k should be 1, i.e. the serial computation. For the present PIC-based solver, it however would not be practical as the memory requirement due to the double grid system for such computation is rather high for a single processor. Thus, k was chosen to be 16 in the present computation. From Fig. 4, it can be seen that the efficiency drops generally from around 0.9 to 0.5 as the number of processors increases from 32 ($p/k = 2$) to 208 ($p/k = 13$). This implies that the computational costs for non-parallel operations such as data communication, which is also related to the interconnection of processors, become

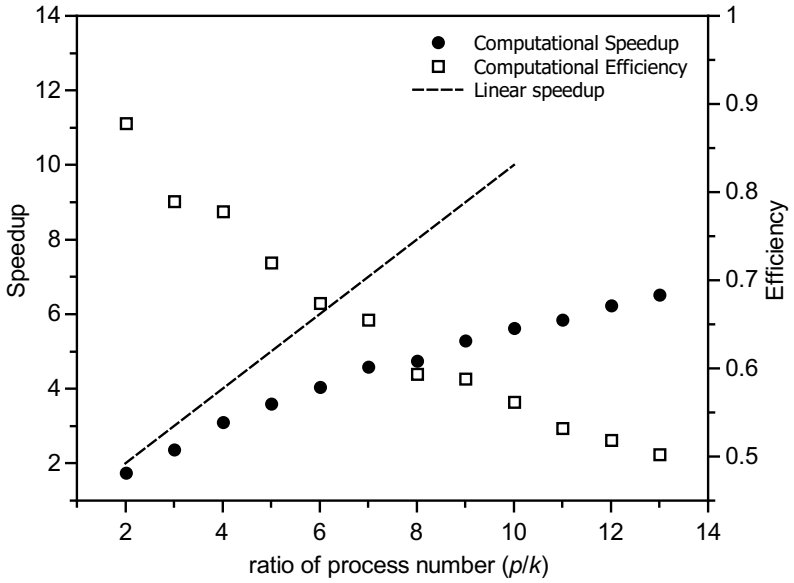

Fig. 4: Speedup and efficiency trends. $k = 16$.

more prominent as a larger number of processors are used. The decrease in computational efficiency is also reflected in the sub-linear speed-up trend. Nevertheless, it can be seen that when the number of processors is relatively small, e.g. around $p/k = 3$, the speedup is close to linear. To further improve the parallel performance, a load balancing scheme could be implemented. The idea being that the size of the Eulerian computational domain for each processor can be changed based on the number of particles they have during the simulation, such that the number of particles in each processor is approximately equal.

3. Test cases

In this section, a number of test cases are presented in which the 3D parallel PICIN model is validated against experimental data as well as results from other state-of-the-art VOF-based numerical models. Attention has been focused on the validation of the model for water wave interactions with vertical cylindrical structures. Two different test cases are selected: i) regular and focused wave interaction with a single cylinder (Chen et al., 2014); ii) solitary wave interaction with single and multiple cylinders (Mo and Liu, 2009). All the test cases were run on the HPCS at the University of Bath, UK.

Table 1: Experimental wave parameters used in this study.

Test conditions	R	F1	F2	F3
T (s)	1.22	1.22	1.63	1.22
f (Hz)	0.82	0.82	0.61	0.82
A (m)	0.07	0.07	0.12	0.11
kh	1.39	1.39	0.86	1.39
ka	0.37	0.37	0.25	0.37
kA	0.2	0.2	0.2	0.3

3.1. Regular and focused wave interaction with a single cylinder

3.1.1. Model setup

This test case concerns regular and focused wave interaction with a single cylinder, typically encountered in offshore wind turbine foundations. This test case was experimentally studied by Zang et al. (2010) and numerically simulated using OpenFOAM® by Chen et al. (2014). The experiment was carried out in the shallow water basin (35 m × 25 m) at the Danish Hydraulic Institute (DHI), where waves are generated with a segmented piston paddle array installed at one end of the basin. The water depth, h, was set to 0.505 m. The cylinder, with a diameter of 0.25 m, was placed in the middle of the basin, with the distance between the upstream stagnation point of the cylinder and the wavemaker being 7.52 m. The main points of interest of this test case are the wave elevation around the cylinder, wave run-up and the horizontal wave force on the cylinder.

Four different wave conditions were investigated, and the parameters for these waves are given in Table 1, where "R" represents regular wave and "F1", "F2" and "F3" represent focused waves; T is the wave period; f is the wave frequency; A represents the wave amplitude; k is the wave number; a is the radius of the cylinder. Two different slenderness (ka) and two different wave steepness (kA) are used for the focused waves. Note that, for the focused waves, the parameters belong to the peak frequency wave component. Also, for the wave conditions used here, the Keulegan-Carpenter number, $KC = \pi A/a$, is approximately 3.0 for the largest wave amplitude, although the validity of the KC number remains to be investigated for focused waves. This indicates that the studied cases are in an inertia-dominated regime (Paulsen et al., 2014). Thus, the dominant term in the wave loading on the cylinder is the inertia force. This is also noted by Chen et al. (2014) where the laminar flow model of OpenFOAM-2.1.0 is used; their numerical results are used for comparisons for this test case.

Fig. 5 shows a schematic of the setup of the NWT. Compared to the physical wave basin,

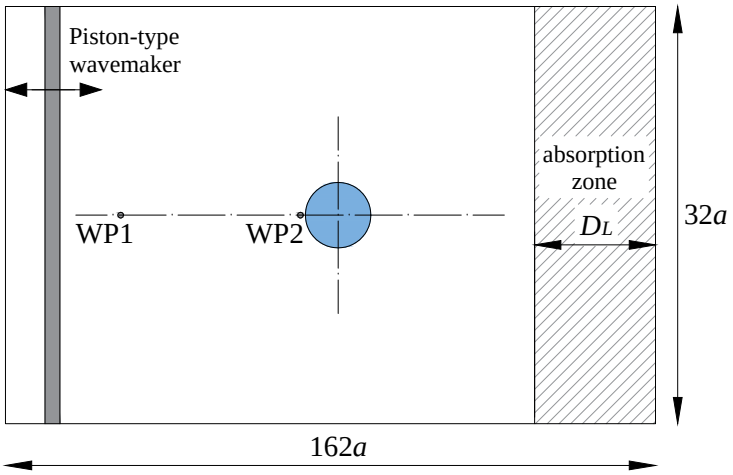

Fig. 5: Schematic (top view) showing the setup of the NWT. a is the radius of the cylinder.

the NWT is reduced to be 20.2 m long, 4 m wide and 0.9 m high, in order to save on CPU cost. The distance between the cylinder and the wavemaker was set the same as that used in the experiment. Wave generation and absorption in the current 3D model use the piston-type wavemaker and the relaxation zone technique described in Section 2.5. In particular, for the focused wave generation, 60 wave components were used and their amplitudes were computed following Chen et al. (2014). The length of the absorption zone (D_L) at the end of the basin was set to 4.7 m and 8.7 m for the regular wave and focused wave cases, respectively. The time histories of wave elevations at two locations were measured: the first wave probe (WP1) was placed at 0.77 m from the wavemaker; the second wave probe (WP2) was located 0.02 m in front of the cylinder.

A grid refinement study on the focused wave generation for F2, in the absence of the cylinder, was used to determine the grid size. Three different grid sizes were chosen: $\Delta x = \Delta y = \Delta z = 0.01$ m (the fine grid), 0.02 m (the moderate grid) and 0.04 m (the coarse grid). Here the width of the NWT was further reduced to 0.4 m to speed up this convergence test, as the wave profiles in the spanwise direction are constant. Fig. 6 plots time histories of the generated focused wave F2 by the three grid configurations at both WP1 and WP2. The results are also compared with the experimental data (the circles). It can be seen that the predicted wave profiles at WP1 are almost indistinguishable, suggesting that the wave characteristics are generated independently from the grid refinement. In addition, the predicted wave profiles match well with the experiment, indicating that the waves are generated correctly. At WP2, however, the wave profile produced by a coarser grid tends

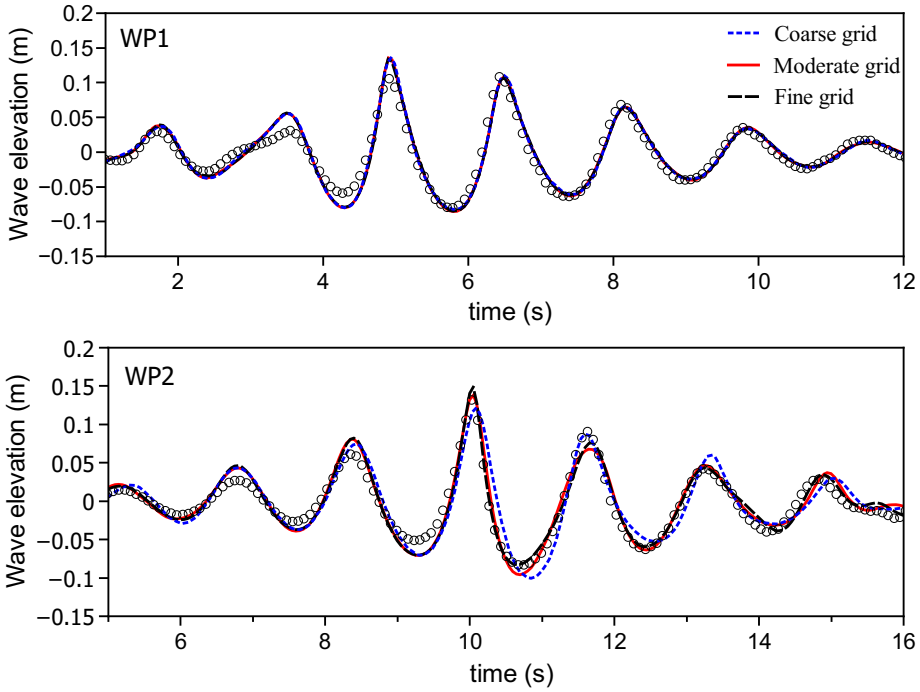

Fig. 6: Grid refinement study of focused wave generation for F2. The circles represent the experimental data.

to have a larger discrepancy, when compared with that of the fine grid; this is caused by the faster accumulation of dispersion and diffusion errors, resulting from a coarser grid, over time as the wave propagates further inside the NWT. Nevertheless, the model is nearly convergent when using the moderate grid (0.02 m), and the result matches well with the experimental data. Considering the increased memory requirement of using the fine grid, the moderate grid is thus selected, resulting in approximately 9.2 million cells and 40.12 million particles, in the presence of the cylinder. The Courant number used is 0.5, which is the same as that used in Chen et al. (2014). The CPU time for the regular wave case is given below, in comparison with that of the OpenFOAM® model of Chen et al. (2014).

3.1.2. Results: comparisons with experiment and OpenFOAM® model

Comparisons for the regular wave case. Fig. 7 shows snapshots of free-surface elevation around the cylinder. It is seen that as the crest of incident wave approaches the cylinder, it causes wave run-up and subsequent run-down of water on the cylinder. The scattered waves due to the cylinder are mostly of higher frequency, as the wave lengths are visibly shorter than that of the incident wave. Furthermore, the scattered waves interact with each other, as well as the incident wave, resulting to a steep "rooster tail" shaped wave at the lee side of the cylinder, which is very similar to the numerical result presented in Kim

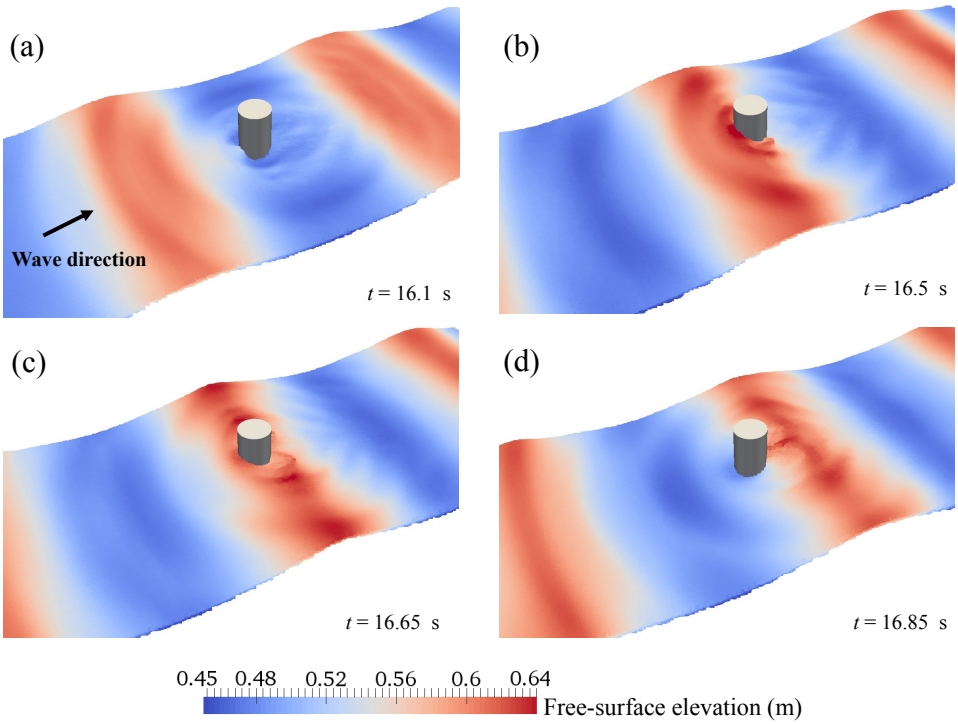

Fig. 7: Snapshots of the numerical simulation for regular wave (R) interaction with a single cylinder.

et al. (2006).

Fig. 8 presents quantitative comparisons between numerical predictions and experimental measurements for the steady state results, including wave elevations at WP1 and WP2 and the horizontal wave force on the cylinder. The corresponding amplitude spectra calculated by a fast Fourier transform (FFT) algorithm are also plotted for further comparison. It can be seen that, in general, results from both numerical models agree well with the experimental measurements, and the present PICIN model produces very similar results to those from the OpenFOAM® model of Chen et al. (2014). In addition, from the comparisons of the spectra it is demonstrated that the present 3D model simulates the nonlinear effect of the wave dynamics well.

Table 2 gives a comparison of the CPU time between the present model and the OpenFOAM® model. It is noted that this comparison is not strictly consistent in a few points: i) PICIN is currently a single-phase solver, whilst the OpenFOAM® (interFoam) model is two-phase; ii) numerical schemes of free-surface capturing and solution of the advection term, and the grid configuration are different; iii) the simulations were run on different numbers of cores. Nevertheless, as can be seen from the magnified CPU cost, this rough comparison demonstrates that, for this type of modelling, PICIN is only slightly less efficient (in terms of CPU cost)

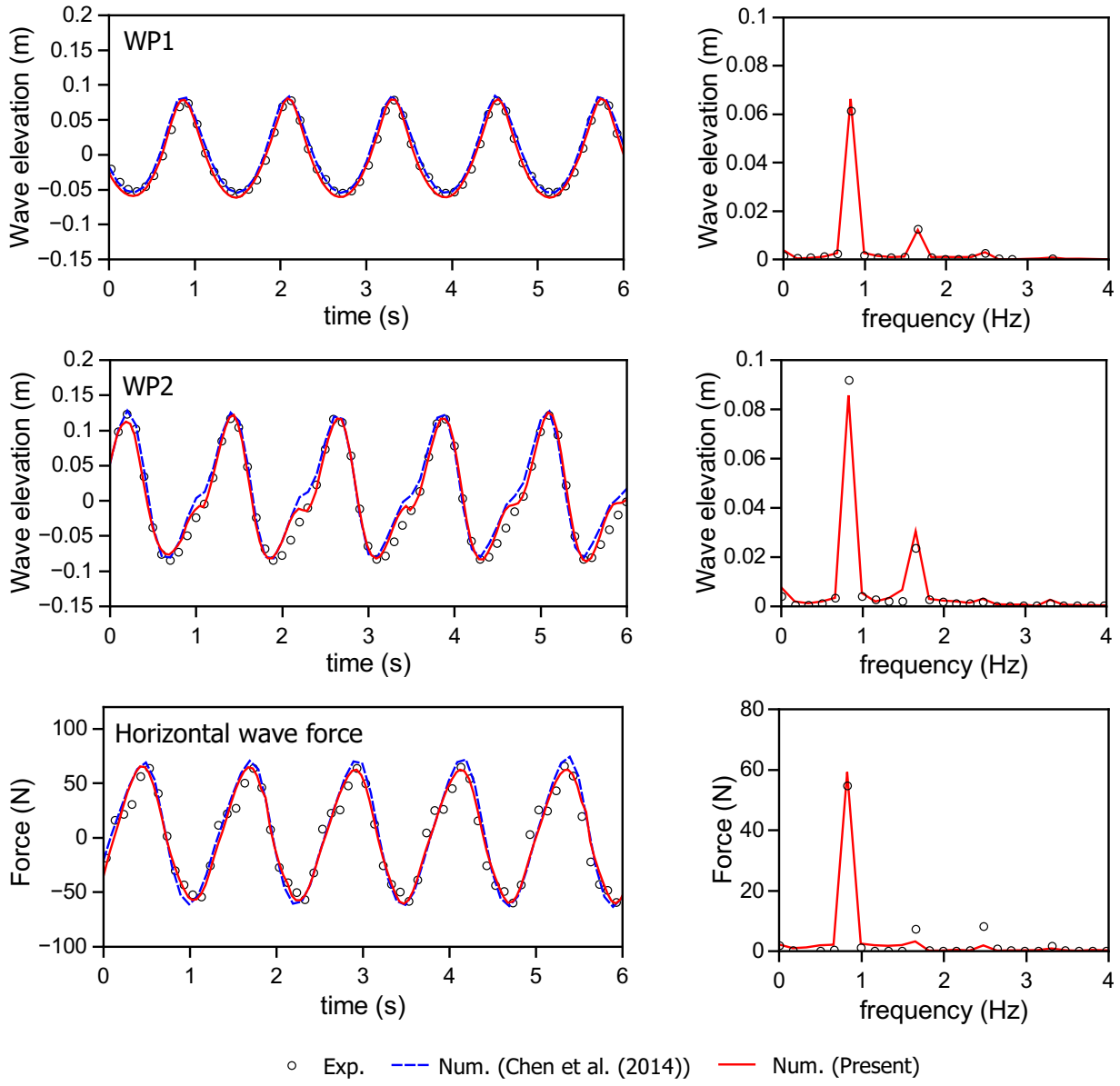

Fig. 8: Comparisons of wave elevations at WP1 and WP2, the horizontal wave force and their corresponding spectra for the regular wave case ($ka = 0.37$; $kA = 0.2$).

than OpenFOAM®, which is one of the widely used CFD models in coastal and offshore engineering applications (see e.g. Higuera et al. (2013)).

Comparisons for the focused wave cases. In order to generate similar focused waves as those used in the physical experiment, the experimental measurements of wave profiles at WP1 were used to calibrate the present numerical model; both the experimental and the numerical wave elevations, as well as their corresponding amplitude spectra, are

Table 2: Comparison of CPU time between the PIC-based PICIN model and the OpenFOAM® model of Chen et al. (2014) for the regular wave case.

Items	PIC-based PICIN	OpenFOAM®
Cell number n (million)	9.20	8.33
Simulated time t_s (T)	18.0	24.5
Cores p	80	8
Total CPU time C_t (h)	12.16	126.20
Magnified CPU cost: $\frac{p \times C_t}{n \times t_s}$	5.87	4.95

plotted in Fig. 9 for comparison. It can be seen that the numerical results of both wave elevations and wave energy distributions agree well with the experimental data for all the wave conditions. These ensure that the incident waves are well reproduced in the simulation, for the investigation of wave-structure interaction.

Fig. 10 and Fig. 11 present comparisons of wave elevations at WP2 and horizontal wave forces on the cylinder, respectively. Numerical results from the PICIN model and the OpenFOAM® model of Chen et al. (2014) (where only data for cases F1 and F2 are available) are plotted against the experimental data. It is seen that both numerical models achieve good agreements with the experiment. In particular, for the case F2, where the slenderness ka is smaller than that in the case F1, the wave elevation and the horizontal wave force tend to be more nonlinear, which is confirmed from the spectra as the ratio of high order harmonics to the first harmonic appears to be larger in the case F2. It seems that, for this case, the present model better captures the energy distributions around the second-order harmonics. Furthermore, for the case F3, where the wave steepness is larger ($kA = 0.3$) and hence the waves are highly nonlinear and close to the breaking point, the present model also exhibits very good agreement with the experiment for both the wave elevation and the wave loading on the cylinder. This demonstrates that, in the present PIC-based PICIN model, the use of Lagrangian particles for solving the nonlinear advection term and tracking the free surface is capable of simulating highly nonlinear free-surface waves, as well as their interaction with structures. It is also noted that the so-called secondary load cycle, that may be associated with a local vortex formed behind the cylinder (Paulsen et al., 2014), is not significant in the current data shown in Fig. 11. It is deduced that this may be because the strength of such vortex is relatively weak. Therefore, together with the small KC number mentioned above, the lack of a turbulence model in the current numerical model seems acceptable for this test case.

It is observed from Fig. 10 that, particularly for cases F2 and F3 where nonlinearity

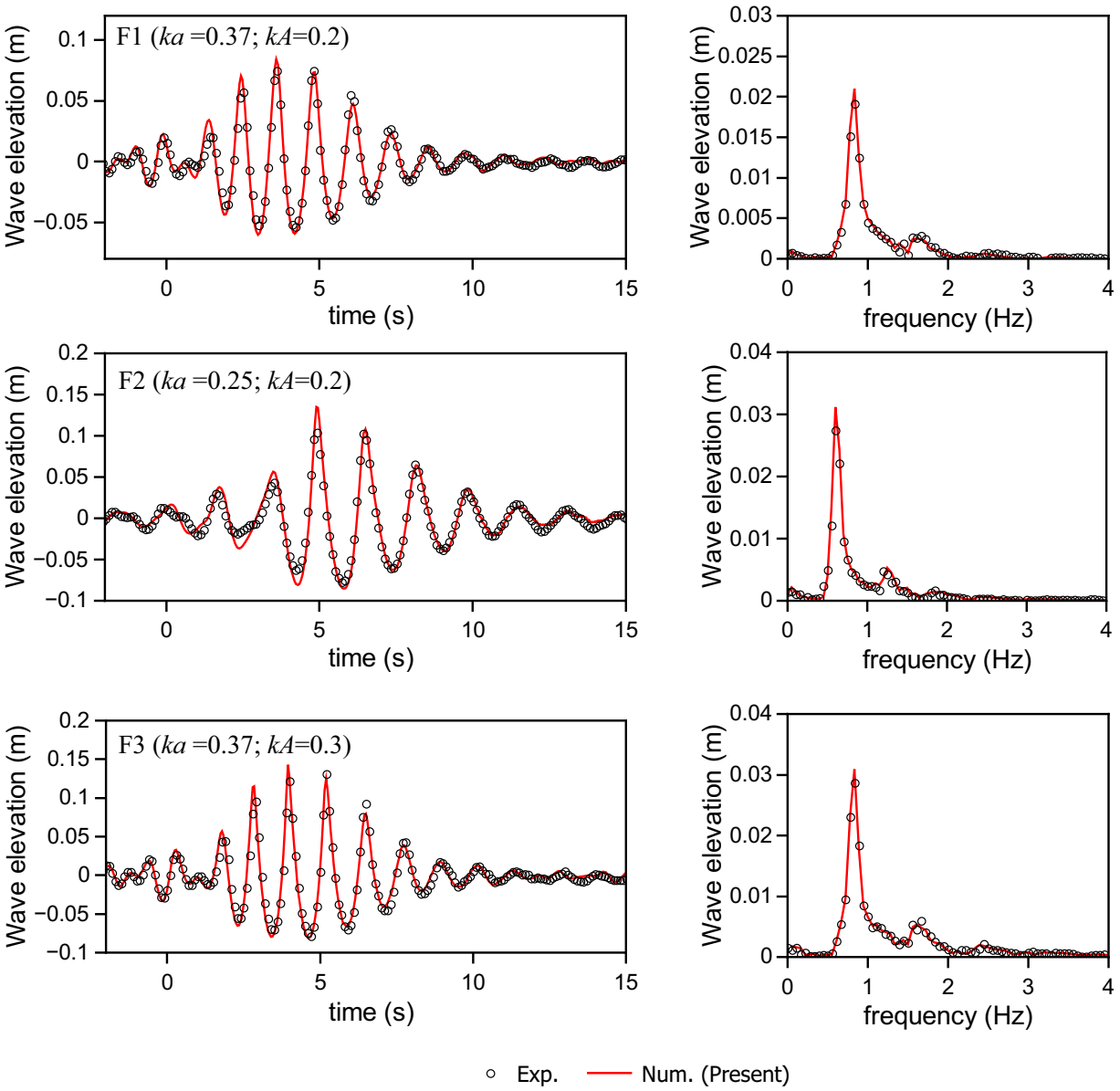

Fig. 9: Comparisons of focused wave elevations at WP1 (left panels) and their corresponding amplitude spectra (right panels).

is strong, small step-shaped waves tend to occur after the focused wave peaks pass the cylinder; for example, around $t = 10.85$ s in the wave elevation of the case F2. This can be explained through snapshots of the present modelling as shown in Fig. 12, from which it can be seen that, at the stagnation point in front of the cylinder, the wave run-up and subsequent strong wash-down of water on the cylinder causes an oscillation of the water surface in front of the cylinder and thus forms the step-shaped waves. These waves are very

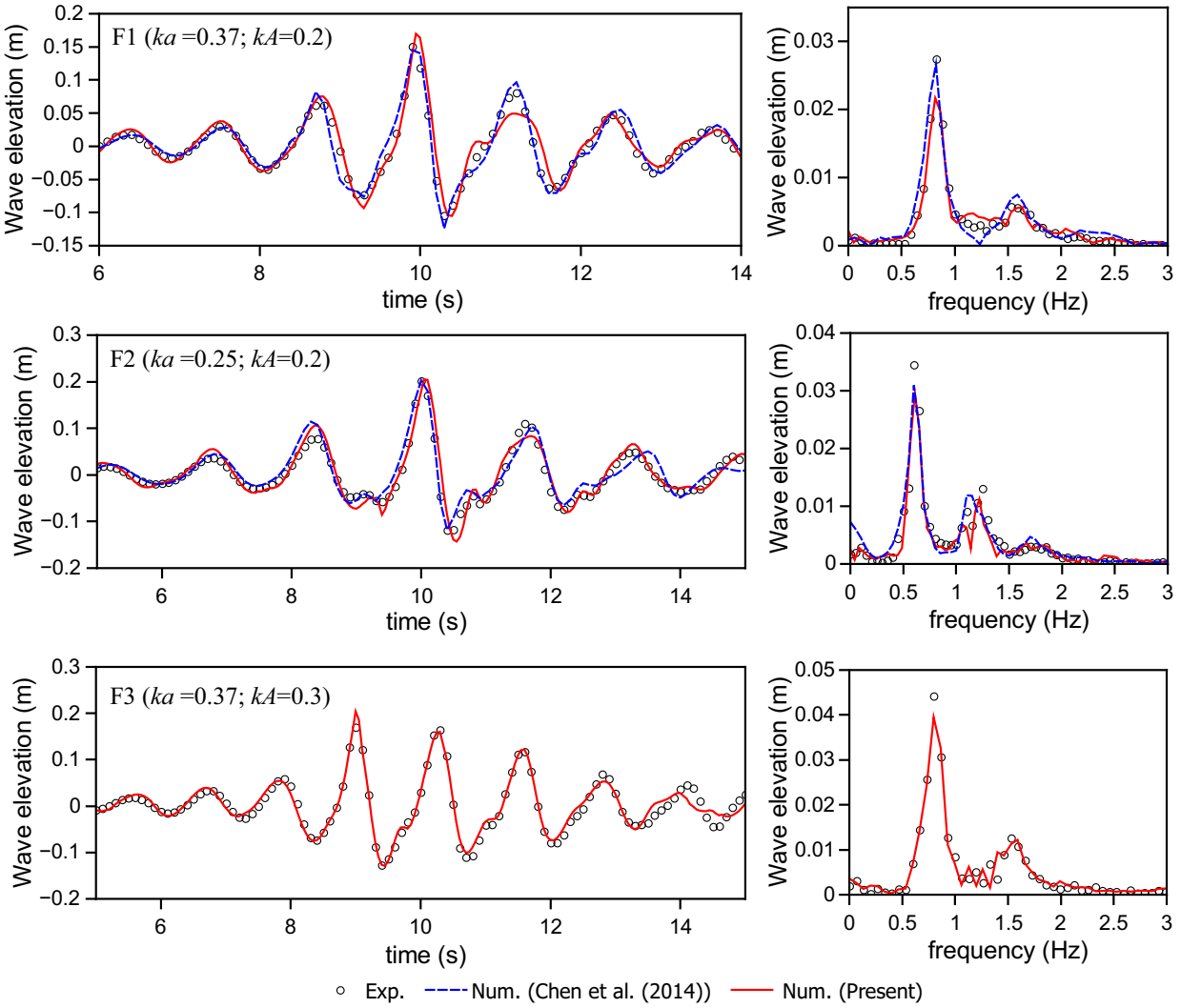

Fig. 10: Comparisons of focused wave elevations at WP2 (left panels) and their corresponding amplitude spectra (right panels).

similar to those predicted by the OpenFOAM® model (c.f. Fig. 14 of Chen et al. (2014)). It is also observed that, after the main wave crest hits the cylinder, the wave shapes formed at the lee side of the cylinder are similar to those shown in the regular wave case.

3.2. Solitary wave interaction with a group of vertical cylinders

3.2.1. Model setup

This test case demonstrates the capability of the parallel PICIN model to cope with solitary wave interaction with a group of vertical cylinders. This case was numerically and experimentally studied by Mo and Liu (2009), where both laboratory data and numerical results are available for comparison. In Mo and Liu (2009), the numerical model uses a

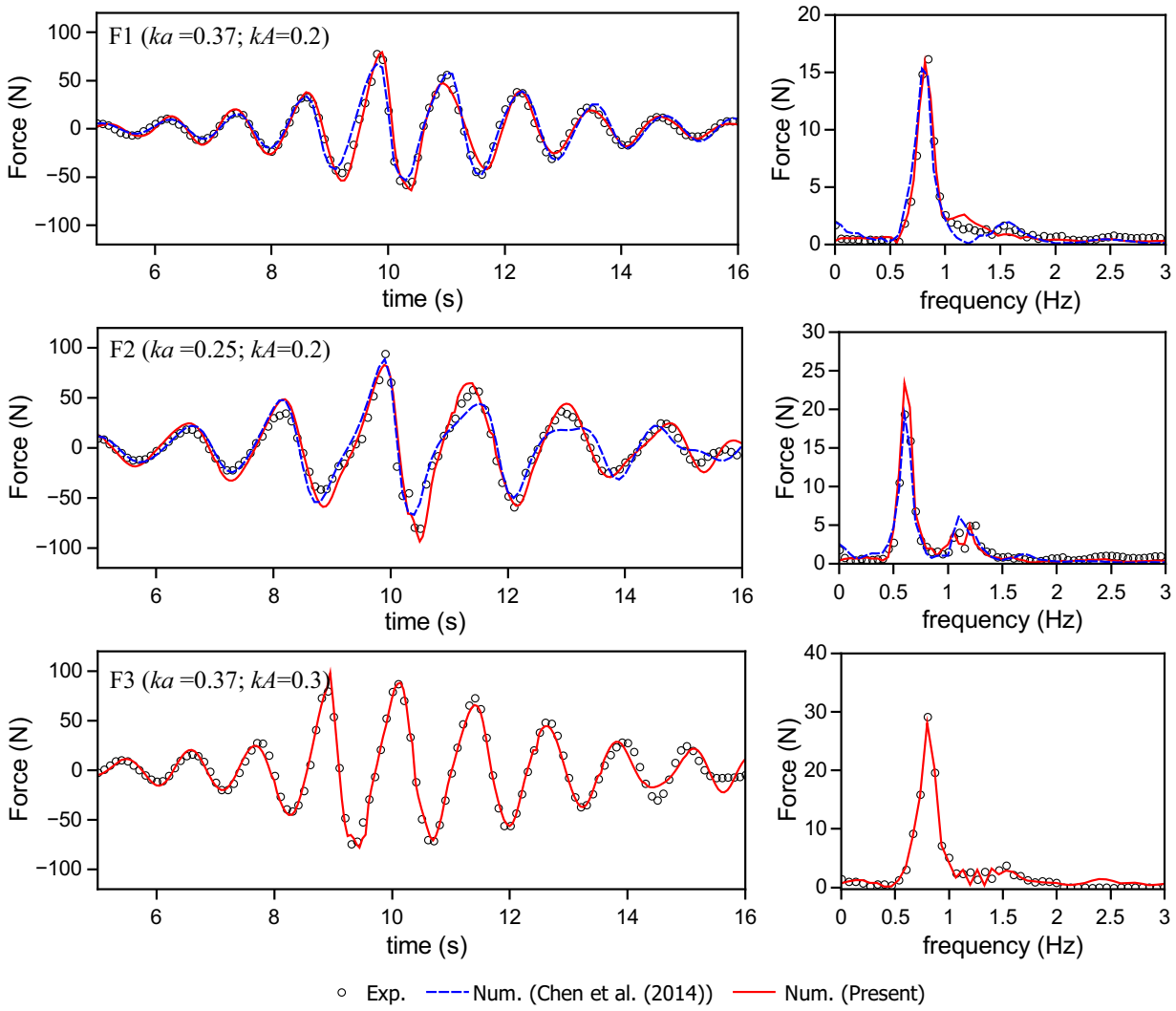

Fig. 11: Comparisons of the horizontal wave forces on the cylinder (left panels) and their corresponding amplitude spectra (right panels).

finite volume scheme coupled with the VOF method for free-surface capturing. However, it is noted that their solver is not based on the OpenFOAM® suite. Moreover, in that paper, the Euler's equations are solved without any dissipative mechanism since the investigated solitary waves are non-breaking. Their experiments were conducted in the Tsunami Wave Basin at the O. H. Hinsdale Wave Research Laboratory (WRL) at Oregon State University (OSU).

In the experiment, either one or three identical stainless steel circular cylinders with a diameter, D, of 1.219 m were instrumented and installed in the basin. Fig. 13 shows a sketch of the cylinder locations. The cylinders were symmetrically placed in the basin, with the

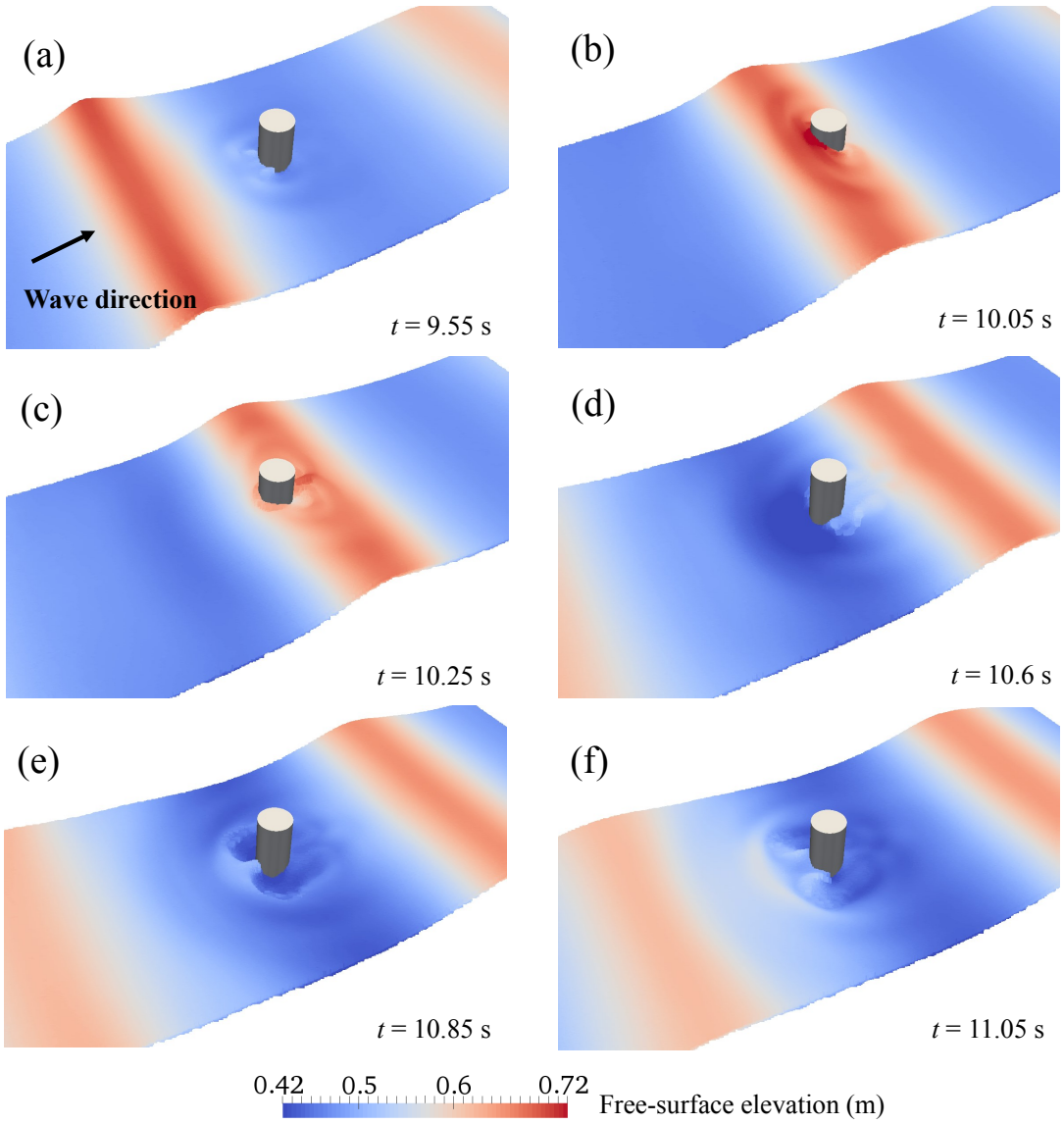

Fig. 12: Snapshots of the numerical simulation for focused wave (F2) interaction with a single cylinder.

centre-to-centre distances being $2D$ between the adjacent cylinders in both the streamwise and the spanwise directions. The single cylinder at the downstream side is hereafter referred to as the middle cylinder, and the other two front cylinders are named as the front cylinders. Two test conditions in this section were used: tests of only one cylinder, i.e. the middle cylinder, and all three cylinders. In both the single and the multiple cylinders cases, the water depth, h, was fixed at 0.75 m and the non-breaking solitary wave with a wave height, H, of 0.3 m was used, such that the relative wave height $H/h = 0.4$. Wave gauges were used to measure free-surface elevations around the cylinders, and locations of the two wave

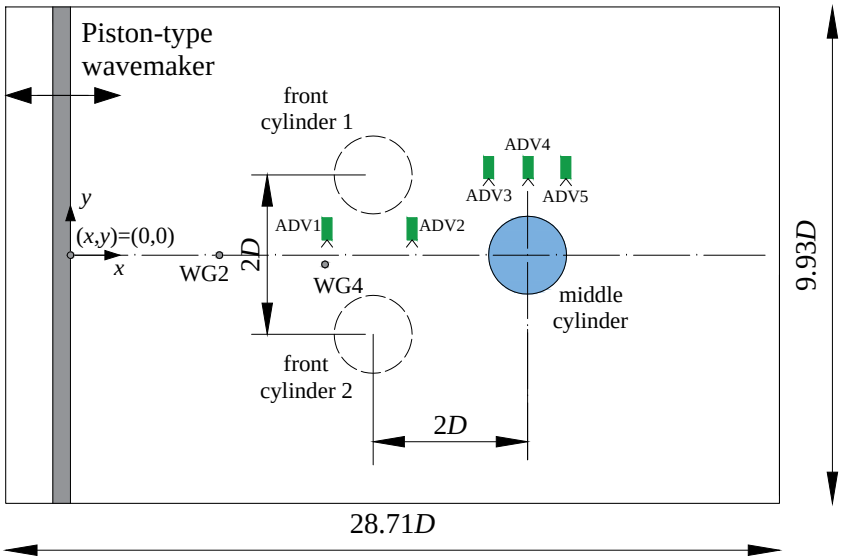

Fig. 13: Schematic (top view) showing the locations of three identical cylinders. D is the diameter of the cylinder.

gauges, WG2 and WG4, used in this study are sketched in Fig. 13. Five Acoustic Doppler Velocimetries (ADVs) were deployed to measure the fluid velocity around the cylinders and their locations are also sketched in Fig. 13. In addition, pressure transducers were installed on the middle cylinder to measure the fluid pressure. The measurements from 4 pressure transducers installed along the frontline of the middle cylinder are used for comparisons in this paper.

For the numerical modelling, the NWT was 35.0 m long and 12.1 m wide, similar to those dimensions used in Mo and Liu (2009). The origin of the coordinate system is located at the downstream face of the piston-type wavemaker, at the middle of the NWT, as shown in Fig. 13, with $z = 0$ m indicating the bottom of the tank. The coordinates (in metres) of the centre of the middle cylinder was (17.925, 0), and the wave gauges WG2 and WG4 are located at (11.299, 0) and (14.877, -0.140), respectively. Furthermore, the locations of the 5 ADVs from ADV1 to ADV5 are at (14.890, 0.147, 0.125), (16.109, 0.140, 0.137), (17.328, 0.875, 0.135), (17.893, 0.855, 0.145) and (18.484, 0.830, 0.140). Note that the locations of the cylinders, wave gauges and ADVs in the numerical modelling were shifted consistently to be closer to the wavemaker than those in the experiment to save on CPU cost. Also, for this test case, the absorption zone at the downstream end of the NWT is not used. This is because, according to the formula used in Yates and Wang (1994), the horizontal length

scale over which the solitary wave has significant elevation is estimated to be 8.60 m for this test case; the spatial configuration of the present NWT thus ensures a time scale that is large enough to collect data for comparison with experiment, without being contaminated by the reflected waves from either the wavemaker or the downstream end of the NWT.

The grid size was $\Delta x = \Delta y = \Delta z = 0.03$ m, resulting in, for example, approximately 31.37 million cells with 90.83 million particles (note the height of NWT was 2.0 m) for the three cylinders case. The Courant number was 0.5. It took about 10.5 h for $t(g/h)^{1/2} = 47$ of (nondimensional) simulation time with 80 cores at the University of Bath HPCS for the three cylinders case.

3.2.2. Results: comparisons with experiment and the VOF-based finite volume solver

The solitary wave was then reproduced to interact with the cylinders. Fig. 14 shows snapshots of the current simulations for both the single and three cylinders cases at the same time instants. In particular, Fig. 14(B1) is plotted at the time instant that maximum run-up occurs at the middle cylinder. For the single cylinder case, it can be seen that as the solitary wave passes the cylinder, it causes wave run-up at the front side, and the water level rise at the lee side due to the symmetric collision between the scattered waves, which then washes frontward down the cylinder. For the three cylinders case, similar processes can be observed for each individual cylinder. It is seen from Fig. 14(B3) that a local wave breaking occurs between the two front cylinders. This is due to an interaction of the symmetrical scattered waves from the front two cylinders as shown in Fig. 14(B2), combined with the reflected waves from the middle cylinder. This local wave breaking is also noted in Mo (2010). However, it is argued that this relatively small scale local wave breaking may not be able to significantly influence the whole fluid property fields, and as such, turbulence effects due to this local wave breaking may not be predominant in this test case.

The quantitative comparisons for wave elevation, wave run-up at the upstream point of the middle cylinder and wave loading on the middle cylinder are shown in Fig. 15 and Fig. 16. Results from both the single cylinder and the multiple cylinders cases are presented. In particular, Fig. 15 shows the comparisons of the non-dimensional wave elevations, η/H, at WG4 and WG2 for the single and the three cylinders cases, respectively. This is to calibrate the present model for the incident solitary wave generation. It is seen that the numerical results match well with experimental measurements, indicating that the solitary waves generated by the PICIN model are acceptable for this test case.

Fig. 16 shows the side-by-side comparisons of the non-dimensional wave run-up, R/H, and the non-dimensional horizontal wave force, $F/\rho g D^3$, at the middle cylinder. For the

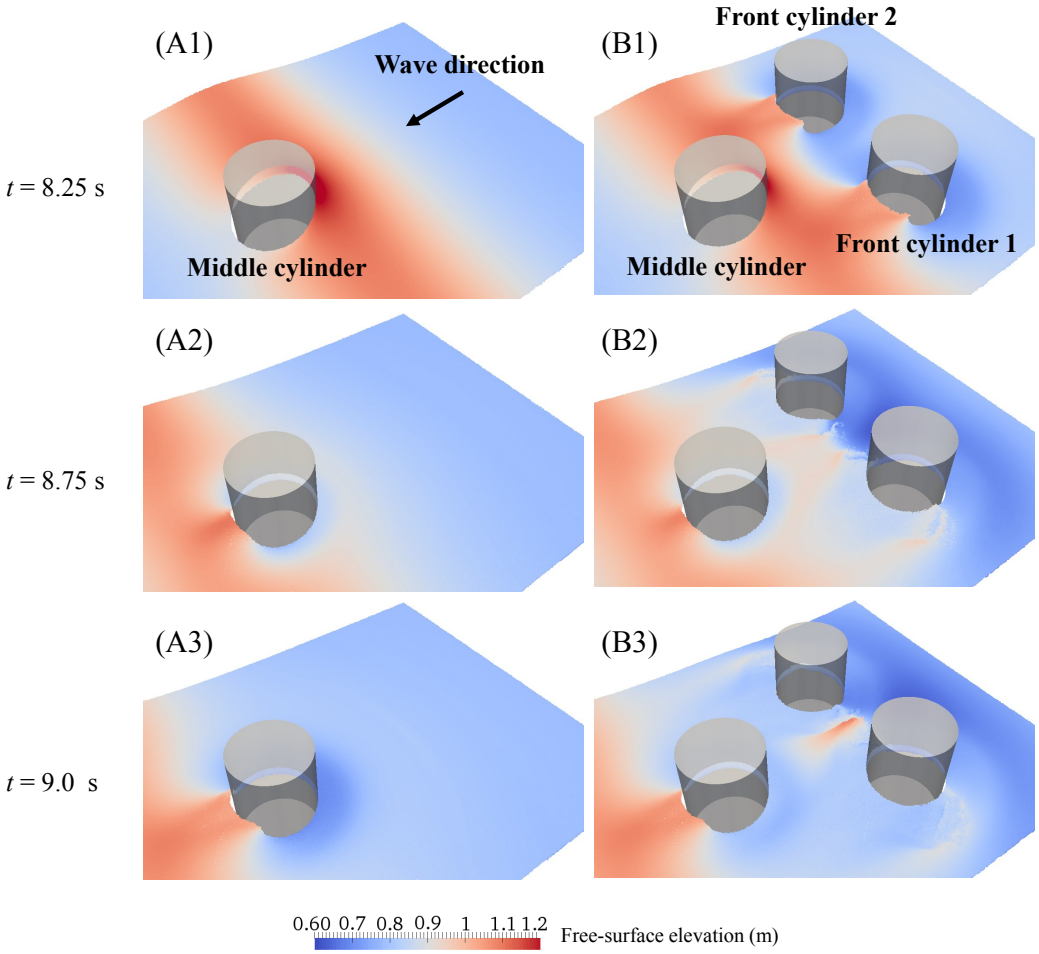

Fig. 14: Snapshots of the numerical simulations for solitary wave interaction with a single cylinder (A1–A3) and multiple cylinders (B1–B3).

wave run-up, it is seen that, in the three cylinders case, the present result is clearly an improvement on that obtained using the VOF-based solver of Mo and Liu (2009). This hints that the present model correctly simulates the scattered waves from the two front cylinders while they interact with each other and propagate to the front point of the middle cylinder. For the horizontal wave force, it may be seen that generally both numerical models predict similar results. A slight reduction of the peak wave force for the three cylinders case, compared with the single cylinder case, is shown in both the numerical predictions and experimental measurements. This is likely a result of the blockage of the front two cylinders. For the single cylinder case, both numerical models slightly underestimate the peak wave force, as well as the trough wave force. The reason for this, however, is still unclear. In general, the numerical results are in good agreement with the experimental data. We note

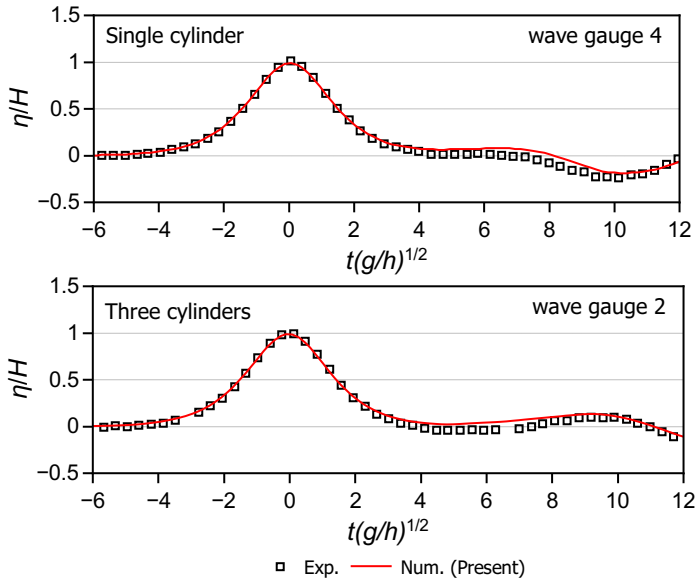

Fig. 15: Comparisons of wave elevations.

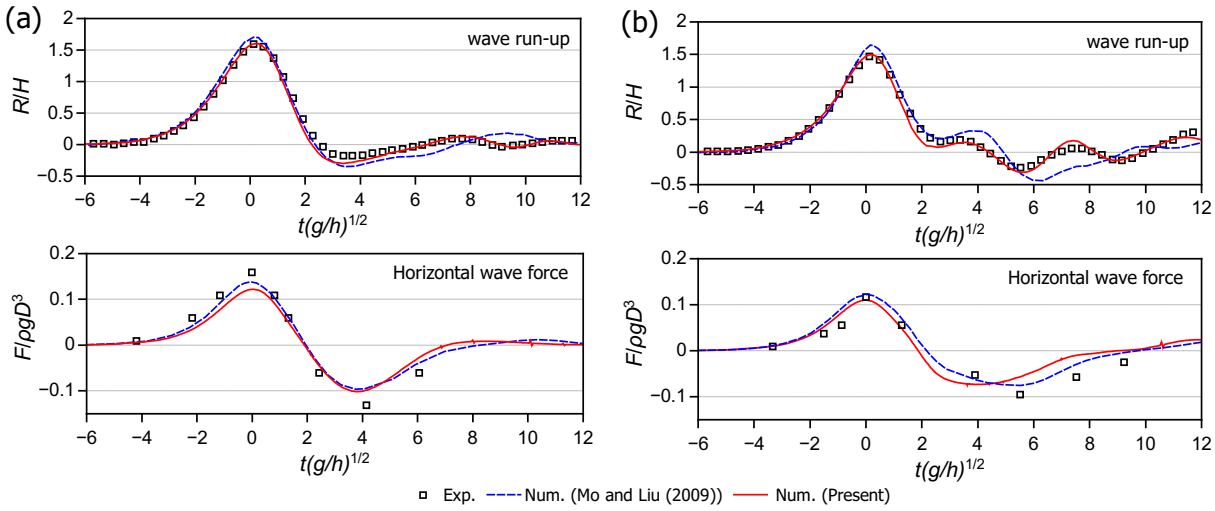

Fig. 16: Comparisons of wave run-up and horizontal wave force at the middle cylinder between experimental and numerical results: (a) single cylinder case; (b) three cylinders case.

that the experimental data are digitised from Mo and Liu (2009) without selection.

Fig. 17 compares the wave elevations at WG4 in the three cylinders case between the PICIN model and the VOF-based finite volume model of Mo and Liu (2009). This is particularly interesting as the location of WG4 is close to the position at which the local wave breaking occurs as shown in Fig. 14(B3). It can be seen from Fig. 17 that the present numerical result matches very well with the experimental measurement. In particular, around

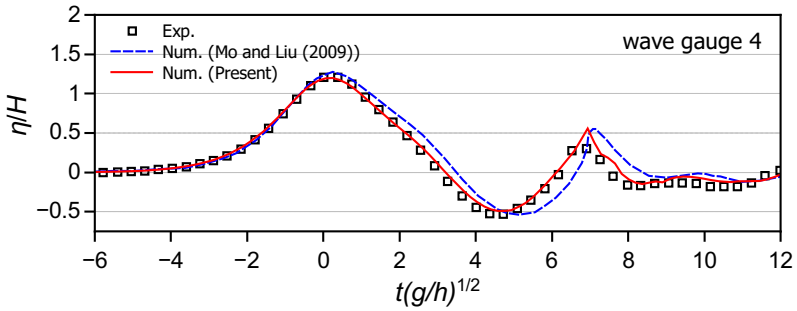

Fig. 17: Comparison of the free-surface elevation at wave gauge 4 in the three cylinders case.

the second wave peak that is effected by the local wave breaking, the present model appears to perform better. This demonstrates that in PICIN the use of particles for tracking the free surface is capable of handling large free-surface deformations such as wave breaking, both straightforwardly and accurately.

Fig. 18 shows the comparisons of the time histories of the flow velocities at different ADVs for the three cylinders case. From Fig. 18, it can be seen that in general, the comparisons between the predicted flow velocities by PICIN and the experimental measurements show good agreement. Some of the current results appear to be better than those obtained by Mo and Liu (2009) when compared to the experimental data; see, e.g., ADV3.

Fig. 19 shows the comparisons of the non-dimensional fluid dynamic pressure, $P/\rho gH$, at different depths, Z, above the bed on the frontline of the middle cylinder for the three cylinders case. From Fig. 19, it is seen that the numerical results from PICIN and the Eulerian model of Mo and Liu (2009) match well with each other. On the other hand, the comparisons between the numerical and the experimental results for $Z/h = 1.0667$, 0.5333 and 0.2667 are disappointing, although the agreement for $Z/h = 0.1333$ is reasonably good. This is probably due to that, as noted in Mo and Liu (2009), some of the pressure transducers may not function properly in the experiment.

4. Conclusions

This paper extends the PIC-based PICIN model (Kelly et al., 2015; Chen et al., 2016b) to three spatial dimensions and parallelises the model using MPI approach. The performance of the 3D parallel PICIN model for free-surface flows and fluid-structure interactions within the coastal and offshore engineering field has been assessed. The tested cases involve: i) regular wave and focused wave interaction with a single cylinder (Chen et al., 2014), and ii) solitary wave of large relative wave height interaction with single or multiple cylinders (Mo and Liu, 2009). For all the validation cases, the 3D PICIN model is shown to be able to well

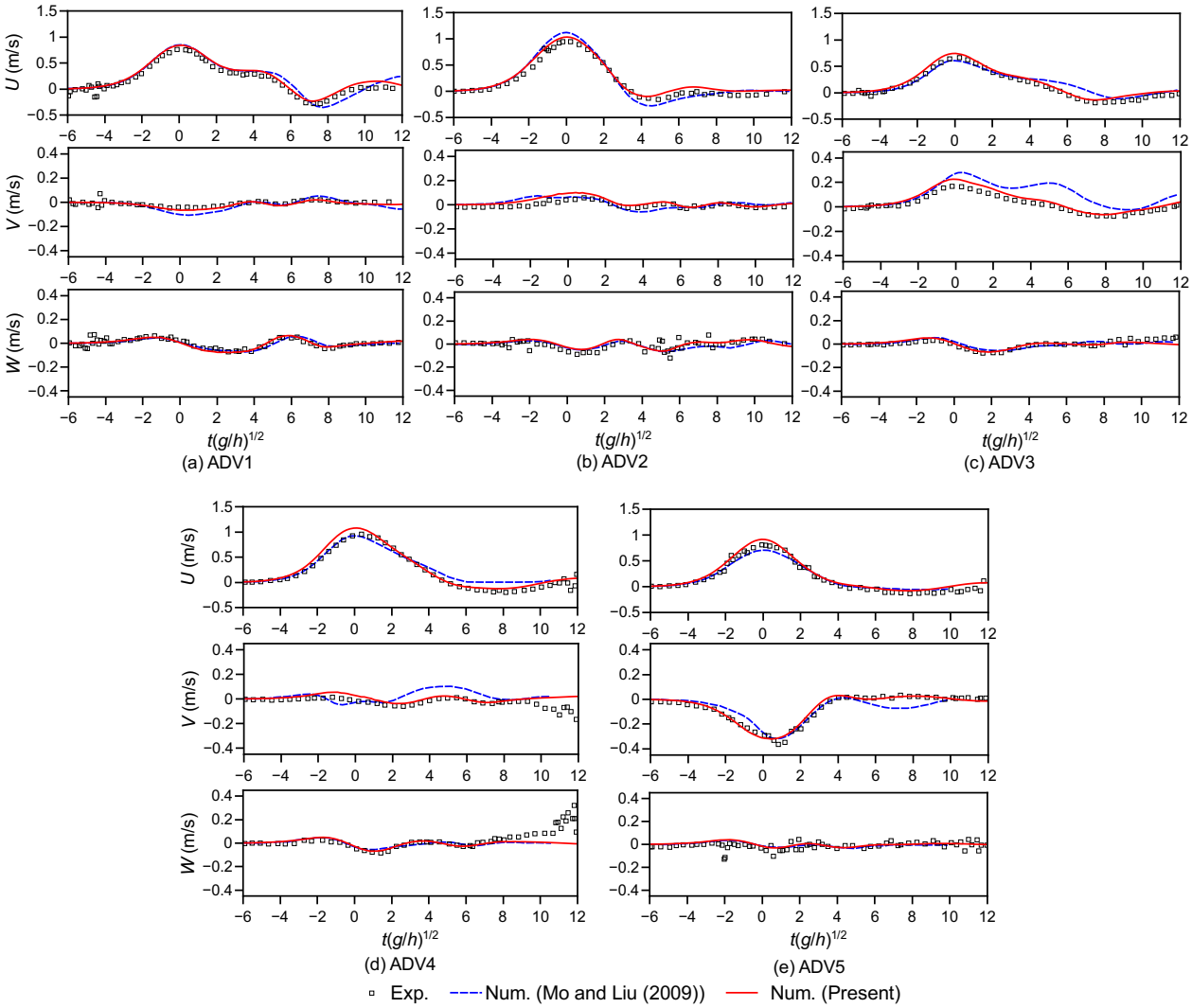

Fig. 18: Comparisons of flow velocities for the numerical and experimental results at different ADVs for the three cylinders case. U, V and W correspond to the flow velocities in the x-, y- and z-directions, respectively.

reproduce the key physical processes. In particular, the highly non-linear water waves and breaking waves, occurred in each of the test cases, are well resolved. This appears to be one of the primary benefits arising from the particle nature of the PIC method in tracking the free surface and solving the transport terms of the governing equations. Thus, the 3D parallel PICIN solver is shown to have the potential flexibilities akin to meshless Lagrangian methods in solving complex water wave problems. In addition, it is shown by a rough comparison that the present PIC model achieves a computational efficiency (in terms of CPU cost) that is similar to the widely used Eulerian model OpenFOAM®. Therefore, this paper illustrates that the PIC-based PICIN model, with new innovations, has great potential to

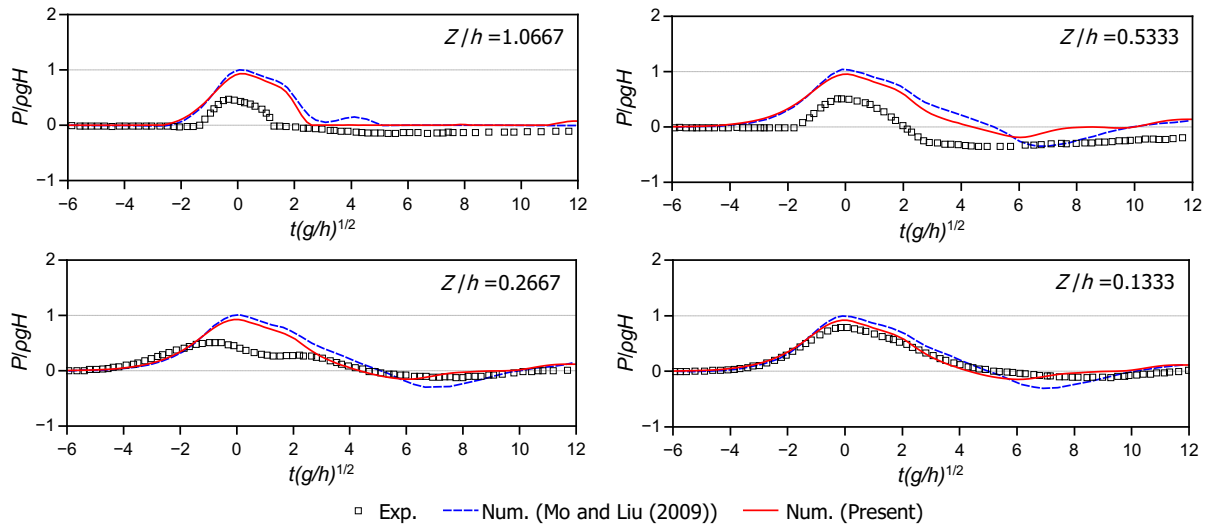

Fig. 19: Comparisons of fluid dynamic pressures at different depths, Z, above the bed on the frontline of the middle cylinder for the three cylinders case.

become a high quality tool for use in coastal and offshore engineering applications. As future improvements, the model would benefit from the inclusion of compressible effects in order to better account for effects like air cushioning (via a two-phase approach), a turbulence model in order to handle the flow during and post wave breaking and a coupled dynamic adaptive grid and particle merging/splitting approach in order to locally minimise truncation errors.

Acknowledgements

This work was funded by the University of Bath (Graduate school funding, sponsor code: 3451) and HR Wallingford (internal research project: DDY0485). This work made use of the Balena High Performance Computing (HPC) Service at the University of Bath. The support of Dr. Giovanni Cuomo, Research Director, HR Wallingford for this work is acknowledged. We also thank Dr. Lifen Chen for providing the numerical data of the OpenFOAM® model. The first author gratefully thanks both institutions for sponsoring his PhD study. The second author thanks all the staff at DHI and the European Community's Sixth Framework Programme through the grant to the budget of the Integrated Infrastructure Initiative HYDRALAB III within the Transnational Access Activities (Contract no. 022441), for supporting the DHI experiment.

References

Adalsteinsson, D., Sethian, J., 1999. The fast construction of extension velocities in level set methods. Journal of Computational Physics 148 (1), 2 – 22.

URL http://www.sciencedirect.com/science/article/pii/S0021999198960909

Brackbill, J. U., Kothe, D. B., Ruppel, H. M., 1988. FLIP: A low-dissipation, Particle-In-Cell method for fluid flow. Computer Physics Communications 48 (1), 25–38.

Brackbill, J. U., Ruppel, H. M., 1986. FLIP: A method for adaptively zoned, Particle-In-Cell calculations of fluid flows in two dimensions. J. Comp. Phys. 65, 314–343.

Chen, L., Zang, J., Hillis, A., Morgan, G., Plummer, A., 2014. Numerical investigation of wave–structure interaction using OpenFOAM. Ocean Engineering 88, 91–109.

Chen, Q., 2017. Development of a full particle pic method for simulating nonlinear wave-structure interaction. Ph.D. thesis, University of Bath.

Chen, Q., Kelly, D. M., Dimakopoulos, A. S., Zang, J., 2016a. Validation of the PICIN solver for 2D coastal flows. Coastal Engineering 112, 87 – 98.
URL http://www.sciencedirect.com/science/article/pii/S0378383916300321

Chen, Q., Kelly, D. M., Spearman, J., Dimakopoulos, A., Zang, J., Williams, C. J. K., 2015. CFD modelling of fall pipe rock dumping using PICIN. In: Proc. Coastal Sediments '15. San Diego, USA.

Chen, Q., Zang, J., Dimakopoulos, A. S., Kelly, D. M., Williams, C. J., 2016b. A Cartesian cut cell based two-way strong fluid-solid coupling algorithm for 2D floating bodies. Journal of Fluids and Structures 62, 252 – 271.
URL http://www.sciencedirect.com/science/article/pii/S0889974616000153

Chorin, A. J., 1968. Numerical solution of the Navier–Stokes equations. Math. Comput. 22, 745–762.

Dalrymple, R. A., Rogers, B. D., 2006. Numerical modeling of water waves with the SPH method. Coastal Engineering 53 (23), 141 – 147.
URL http://www.sciencedirect.com/science/article/pii/S0378383905001304

Dean, R. G., Dalrymple, R. A., 1991. Water wave mechanics for engineers and scientists. Prentice-Hall.

Edwards, E., Bridson, R., 2012. A high-order accurate Particle–In–Cell method. International Journal for Numerical Methods in Engineering 90 (9), 1073–1088.
URL http://dx.doi.org/10.1002/nme.3356

Gibou, F., Fedkiw, R. P., Cheng, L.-T., Kang, M., 2002. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. Journal of Computational Physics 176 (1), 205 – 227.
URL http://www.sciencedirect.com/science/article/pii/S0021999101969773

Harlow, F. H., 1955. A machine calculation method for hydrodynamic problems. Technical Report LAMS-1956, Los Alamos Scientific Laboratory, Los Alamos.

Harlow, F. H., 1964. The Particle–In–Cell computing method for fluid dynamics. In: Alder, B. (Ed.), Methods in Computational Physics. Academic Press, New York, pp. 319–343.

Harlow, F. H., Welch, J. E., 1965. Numerical calculation of time–dependent viscous incompressible flow of fluid with free surface. Physics of Fluids 8, 2182–2189.

Higuera, P., Lara, J. L., Losada, I. J., 2013. Simulating coastal engineering processes with OpenFOAM®. Coastal Engineering 71, 119 – 134.
URL http://www.sciencedirect.com/science/article/pii/S0378383912001093

Jacobsen, N. G., Fuhrman, D. R., Fredse, J., 2012. A wave generation toolbox for the open-source CFD library: OpenFoam®. International Journal for Numerical Methods in Fluids 70 (9), 1073–1088.
URL http://dx.doi.org/10.1002/fld.2726

Jiang, C., Schroeder, C., Selle, A., Teran, J., Stomakhin, A., Jul. 2015. The Affine Particle-in-cell Method. ACM Trans. Graph. 34 (4), 51:1–51:10.
URL http://doi.acm.org/10.1145/2766996

Kamath, A., Chella, M. A., Bihs, H., Arntsen, Ø. A., 2015. Evaluating wave forces on groups of three and nine cylinders using a 3D numerical wave tank. Engineering Applications of Computational Fluid Mechanics 9 (1), 343–354.
URL http://dx.doi.org/10.1080/19942060.2015.1031318

Kelly, D. M., 2012. Full particle PIC modelling of the surf and swash zones. In: Proc. 33rd Int. Conf. Coast. Eng. A.S.C.E., Santander, pp. 77–92.
URL http://dx.doi.org/10.9753/icce.v33.currents.30

Kelly, D. M., Chen, Q., Zang, J., 2015. PICIN: A Particle–In–Cell solver for incompressible free surface flows with two-way fluid-solid coupling. SIAM Journal on Scientific Computing 37 (3), B403–B424.
URL http://dx.doi.org/10.1137/140976911

Kim, J. W., Kyoung, J. H., Ertekin, R. C., Bai, K. J., 2006. Finite-element computation of wave-structure interaction between steep stokes waves and vertical cylinders. Journal of waterway, port, coastal, and ocean engineering 132 (5), 337–347.

Lara, J. L., Higuera, P., Guanche, R., Losada, I. J., 2013. Wave interaction with piled structures: Application with IH-FOAM. In: ASME 2013 32nd International Conference on Ocean, Offshore and Arctic Engineering. American Society of Mechanical Engineers, pp. V007T08A078–V007T08A078.

Leschka, S., Oumeraci, H., 2014. Solitary waves and bores passing three cylinders-effect of distance and arrangement. Coastal Engineering Proceedings 1 (34), 39.

Lind, S. J., Stansby, P. K., Rogers, B. D., 2016. Fixed and moored bodies in steep and breaking waves using sph with the froude–krylov approximation. Journal of Ocean Engineering and Marine Energy 2 (3), 331–354.
URL http://dx.doi.org/10.1007/s40722-016-0056-4

Mo, W., 2010. Numerical investigation of solitary wave interaction with group of cylinders. Ph.D. thesis, Cornell University.

Mo, W., Liu, P. L.-F., 2009. Three dimensional numerical simulations for non-breaking solitary wave interacting with a group of slender vertical cylinders. International Journal of Naval Architecture and Ocean Engineering 1 (1), 20 – 28.
URL http://www.sciencedirect.com/science/article/pii/S209267821630379X

Monaghan, J. J., 1989. On the problem of penetration in particle methods. Journal of Computational Physics 82 (1), 1 – 15.
URL http://www.sciencedirect.com/science/article/pii/0021999189900326

Monaghan, J. J., Lattanzio, J. C., 1985. A refined particle method for astrophysical problems. Astronomy and astrophysics 149, 135–143.

Ng, Y. T., Min, C., Gibou, F., 2009. An efficient fluid–solid coupling algorithm for single–phase flows. Journal of Computational Physics 228 (23), 8807–8829.

Paulsen, B. T., Bredmose, H., Bingham, H. B., Jacobsen, N. G., 2014. Forcing of a bottom-mounted circular cylinder by steep regular water waves at finite depth. Journal of Fluid Mechanics 755, 1–34.

Press, W., Flannery, B., Teukolsky, S., Vetterling, W., 1992. Numerical Recipes: The Art of Scientific

Computing (second edition). Cambridge Univ. Press, New York.

Ralston, A., 1962. Runge–Kutta methods with minimum error bound. Mathematics of Computation 16:80, 431–437.

Wu, N.-J., Hsiao, S.-C., Chen, H.-H., Yang, R.-Y., 2016. The study on solitary waves generated by a piston-type wave maker. Ocean Engineering 117, 114 – 129.
URL http://www.sciencedirect.com/science/article/pii/S0029801816300117

Yates, G. T., Wang, K. H., 1994. Solitary wave scattering by a vertical cylinder: experimental study. In: The Fourth International Offshore and Polar Engineering Conference. International Society of Offshore and Polar Engineers.

Zang, J., Taylor, P. H., Morgan, G., Tello, M., Grice, J., Orszaghova, J., 2010. Experimental study of non-linear wave impact on offshore wind turbine flundations. In: Proceedings of the 3rd International Conference on the Application of Physical Modelling to Port and Coastal Protection. Barcelona, Spain.

Zhao, X., Hu, C., 2012. Numerical and experimental study on a 2-D floating body under extreme wave conditions. Applied Ocean Research 35, 1–13.

Zhou, J. T., Ma, Q. W., Yan, S., Zhang, L., 2009. Numerical investigation of violent wave impact on offshore wind energy structures using MLPG_R method. In: The Nineteenth International Offshore and Polar Engineering Conference. International Society of Offshore and Polar Engineers.

Zhu, S., Moule, G., 1996. An efficient numerical calculation of wave loads on an array of vertical cylinders. Applied Mathematical Modelling 20 (1), 26 – 33.
URL http://www.sciencedirect.com/science/article/pii/0307904X9500108V

Zhu, Y., Bridson, R., 2005. Animating sand as a fluid. ACM Trans. Graph. 24 (3), 965–972.
URL http://doi.acm.org/10.1145/1073204.1073298